

DIGITAL SPREAD SPECTRUM CIRCUITRY

Andrew K. Percey

John D. Logue

F. Erich Goetting

Paul G. Hyland

INSB
7 PRIORITY

8 This is a continuation-in-part of U.S. Patent
9 Application Serial No. 09/102,740 filed June 22, 1998.

99
11 CROSS-REFERENCE TO RELATED APPLICATIONS

12 This application relates to commonly assigned,
13 ~~concurrently filed~~ U.S. Patent Application Serial No.
14 09/102,704 ~~[docket X 440 US]~~, *entitled* "Glitchless Delay Line Using
15 Gray Code Multiplexer", which is incorporated herein by
16 reference.

18 FIELD OF THE INVENTION

19 The present invention relates to delay lock loops (DLLs)
20 for digital electronics. More specifically, the present
21 invention relates to DLLs capable of locking clock signals
22 over a wide frequency range.

24 BACKGROUND OF THE INVENTION

25 Synchronous digital systems, including board level
26 systems and chip level systems, rely on one or more clock
27 signals to synchronize elements across the system.
28 Typically, one or more clock signals are distributed across
29 the system on one or more clock lines. However, due to
30 various problems such as clock buffer delays, high
31 capacitance of heavily loaded clock lines, and propagation
32 delays, the rising edges of a clock signal in different parts
33 of the system may not be synchronized. The time difference
34 between a rising (or falling) edge in one part of the system
35 with the corresponding rising (or falling) edge in another
36 part of the system is referred to as "clock skew".

EL539650663US

Before output clock signal O_CLK reaches logic circuits 190, output clock signal O_CLK is skewed by clock skew 180. Clock skew 180 can be caused by delays in various clock buffers (not shown) or propagation delays on the clock signal line carrying output clock signal O_CLK (e.g., due to heavy loading on the clock signal line). To distinguish output clock signal O_CLK from the skewed version of output clock signal O_CLK, the skewed version is referred to as skewed clock signal S_CLK. Skewed clock signal S_CLK drives the clock input terminals (not shown) of the clocked circuits

1 within logic circuits 190. Skewed clock signal S_CLK is also
2 routed back to delay lock loop 100 on a feedback path 170.
3 Typically, feedback path 170 is dedicated specifically to
4 routing skewed clock signal S_CLK to delay lock loop 110.
5 Therefore, any propagation delay on feedback path 170 is
6 minimal and causes only negligible skewing.

7 Figure 2A provides a timing diagram of reference clock
8 signal REF_CLK, output clock signal O_CLK, and skewed clock
9 signal S_CLK. All three clock signals have the same
10 frequency F (not shown) and period P, and all are active-high
11 (i.e., the rising edge is the active edge). Since output
12 clock signal O_CLK is delayed by propagation delay D, a clock
13 edge 220 of output clock signal O_CLK lags corresponding
14 clock edge 210 of reference clock signal REF_CLK by
15 propagation delay D. Similarly, a clock edge 230 of skewed
16 clock signal S_CLK lags corresponding clock edge 220 of
17 output clock signal O_CLK by a propagation delay SKEW, which
18 is the propagation delay caused by clock skew 180 (Figure 1).
19 Therefore, clock edge 230 of skewed clock signal S_CLK lags
20 clock edge 210 of reference clock signal REF_CLK by a
21 propagation delay DSKEW, which is equal to propagation delay
22 D plus propagation delay SKEW.

23 Delay lock loop 100 controls propagation delay D by
24 controlling delay line 110. However, delay line 110 cannot
25 create negative delay; therefore, clock edge 230 cannot be
26 synchronized to clock edge 210. Fortunately, clock signals
27 are periodic signals. Therefore, delay lock loop 100 can
28 synchronize reference clock signal REF_CLK and skewed clock
29 signal S_CLK by further delaying output clock signal O_CLK
30 such that clock edge 240 of skewed clock signal S_CLK is
31 synchronized with clock edge 210 of reference clock signal
32 REF_CLK. As shown in Figure 2B, propagation delay D is
33 adjusted so that propagation delay DSKEW is equal to period
34 P. Specifically, delay line 110 is tuned so that propagation
35 delay D is increased until propagation delay D equals period
36 P minus propagation delay SKEW. Although propagation delay

1 DSKEW could be increased to any multiple of period P to
2 achieve synchronization, most delay lock loops do not include
3 a delay line capable of creating such a large propagation
4 delay.

5 Phase detector 120 (Figure 1) controls delay line 110 to
6 regulate propagation delay D. The actual control mechanism
7 for delay lock loop 100 can differ. For example, in one
8 version of delay lock loop 100, delay line 110 starts with a
9 propagation delay D equal to minimum propagation delay D_MIN,
10 after power-on or reset. Phase detector 110 then increases
11 propagation delay D until reference clock signal REF_CLK is
12 synchronized with skewed clock signal S_CLK. In another
13 system, delay lock loop 100 starts with a propagation delay D
14 equal to the average of minimum propagation delay D_MIN and
15 maximum propagation delay D_MAX, after power-on or reset.
16 Phase detector 120 then determines whether to increase or
17 decrease (or neither) propagation delay D to synchronize
18 reference clock signal REF_CLK with skewed clock signal
19 S_CLK. For example, phase detector 120 would increase
20 propagation delay D for the clock signals depicted in Figure
21 2A. However, phase detector 120 would decrease propagation
22 delay D for the clock signals depicted in Figure 2C.

23 In Figure 2C, skewed clock signal S_CLK is said to "lag"
24 reference clock signal REF_CLK, because the time between a
25 rising edge of reference clock signal REF_CLK and the next
26 rising edge of skewed clock signal S_CLK is less than the
27 time between a rising edge of skewed clock signal S_CLK and
28 the next rising edge of reference clock signal REF_CLK.
29 However, in Figure 2A, reference clock signal REF_CLK is said
30 to "lag" skewed clock signal S_CLK, because the time between
31 a rising edge of skewed clock signal S_CLK and the next
32 rising edge of reference clock signal REF_CLK is less than
33 the time between a rising edge of reference clock signal
34 REF_CLK and the next rising clock edge of skewed clock signal
35 S_CLK. Alternatively, in Figure 2A skewed clock signal S_CLK
36 could be said to "lead" reference clock signal REF_CLK.

1 After synchronizing reference clock signal REF_CLK and
2 skewed clock signal S_CLK, delay lock loop 100 monitors
3 reference clock signal REF_CLK and skewed clock signal S_CLK
4 and adjusts propagation delay D to maintain synchronization.
5 For example, if propagation delay SKEW increases, perhaps
6 caused by an increase in temperature, delay lock loop 100
7 must decrease propagation delay D to compensate. Conversely,
8 if propagation delay SKEW decreases, perhaps caused by a
9 decrease in temperature, delay lock loop 100 must increase
10 propagation delay D to compensate. The time in which delay
11 lock loop 100 is attempting to first synchronize reference
12 clock signal REF_CLK and skewed clock signal S_CLK, is
13 referred to as lock acquisition. The time in which delay
14 lock loop 100 is attempting to maintain synchronization is
15 referred to as lock maintenance. The value of propagation
16 delay D at the end of lock acquisition, i.e. when
17 synchronization is initially established, is referred to as
18 initial propagation delay ID.

19 However, as explained above, delay line 110 can only
20 provide a propagation delay between a minimum propagation
21 delay D_{MIN} and a maximum propagation delay D_{MAX} . During
22 lock maintenance, delay lock loop 100 may lose
23 synchronization if a propagation delay D smaller than minimum
24 propagation delay D_{MIN} is required to maintain
25 synchronization. Similarly, synchronization may be lost if a
26 propagation delay D greater than maximum propagation delay
27 D_{MAX} is required to maintain synchronization.

For example, if lock acquisition occurs while the system using delay lock loop 100 is at a very high temperature, delay lock loop 100 is likely to achieve synchronization with a very small initial propagation delay ID, since propagation delay SKEW is likely to be large with respect to period P. As the system's temperature increases further, propagation delay SKEW is likely to increase to a point where propagation delay SKEW plus minimum propagation delay D_MIN is greater than period P. In this situation, delay lock loop 100 must

12 In addition, conventional delay lock loop circuits
13 provide for precise synchronization of the reference clock
14 signal REF_CLK and the skew clock signal S_CLK. It would be
15 desirable to have a delay lock loop circuit which is capable
16 of providing a skew clock signal S_CLK which is precisely
17 shifted by a relatively small amount with respect to the
18 reference clock signal REF_CLK. It would further be
19 desirable if such delay lock loop circuit were capable of
20 providing both a leading and lagging relationship. Such a
21 delay lock loop circuit would enable the precise control of
22 clock phase in logic circuits. Such control allows, for
23 example, more accurate timing budget allocation, which in
24 turn, allows synchronous digital systems to run at faster
25 speeds.

2025年12月24日

limits, special techniques must be employed to reach compliance. Conventional compliance techniques include the use of stand-alone (i.e., off-chip) spread spectrum clock oscillators and metal shielding around the radiating components.

It would therefore be desirable to have a clock system that overcomes the electromagnetic emission limitations of delay lock loop 100.

SUMMARY OF THE INVENTION

The present invention provides a delay lock loop that synchronizes the reference clock signal with the skewed clock signal using a delay line having an initial propagation delay within a lock window. The lock window is a period of time between the minimum delay of the propagation delay and the maximum propagation delay. The extent of the lock window is chosen to ensure that changes in environmental conditions or clock frequencies, when compensated for by changing the propagation delay of the delay line, do not cause a loss of synchronization. A delay lock loop in accordance with one embodiment of the present invention incorporates a clock phase shifter in addition to the delay line to synchronize the reference clock. The increased flexibility provided by the clock phase shifter increases the range of frequencies at which the delay lock loop will operate.

The delay line receives the reference clock signal from a reference input terminal of the delay lock loop. The output of the delay line (i.e., the delayed clock signal) is provided to the clock phase shifter, which can generate one or more phase-shifted clock signals. An output generator receives the delayed clock signal and the one or more phase-shifted clock signals. The output generator provides one of the clock signals as the output clock signal on an output terminal. A phase detector compares the reference clock signal with the skewed clock signal, which is received on a feedback input terminal of the delay lock loop, to determine

1 whether to increase or decrease the propagation delay of the
2 delay line to synchronize the reference clock signal and the
3 skewed clock signal.

4 One embodiment of the clock phase shifter generates N-1
5 phase-shifted clock signals. Each of the phase-shifted clock
6 signals is phase-shifted from the other N-2 clock signals and
7 the delayed clock signal by $360/N$ degrees. For example, if
8 the clock phase shifter generated 3 phase-shifted clock
9 signals (i.e., N is equal to four), the phase-shifted clock
10 signals would be phase-shifted from the delayed clock signal
11 by 90 degrees, 180 degrees, and 270 degrees. The clock phase
12 shifter can be implemented using N delay lines and a phase
13 detector.

14 The delay lock loop can include a controller to control
15 the delay line and the output generator. In one embodiment
16 of the invention, the controller causes the output generator
17 to drive the delayed clock signal as the output clock. The
18 controller synchronizes the reference clock signal with the
19 skewed clock signal by adjusting the propagation delay of the
20 delay line to an initial delay. If the initial delay is not
21 within the lock window, the controller causes the output
22 generator to drive a first phase-shifted clock signal as the
23 output signal. The controller and phase detector then
24 synchronize the reference clock signal with the skewed clock
25 signal by adjusting the propagation delay of the delay line
26 to a second initial delay. If the second initial delay is
27 not within the lock window, the controller causes the output
28 generator to use a second phase-shifted clock signal as the
29 output clock. The controller continues in this manner until
30 an initial delay within the lock window is found.

31 In another embodiment of the invention, the clock phase
32 shifter is coupled to receive the reference clock signal.
33 The clock phase shifter generates phase-shifted clock signals
34 that are phase-shifted from the reference clock signals. The
35 reference clock signal or one of the phase-shifted clock
36 signals from the clock shifter is selected to be the input

1 signal of the delay line. The delay line is controlled by
2 the controller and the phase detector to delay the input
3 clock signal and synchronize the skewed clock signal with the
4 reference clock signal.

After the delay lock loop synchronizes the reference clock signal with the skewed clock signal, a digital phase shifter can be used to shift the skewed clock signal by a small amount with respect to the reference clock signal. In accordance with one embodiment, the tap settings and the finer trim settings of a delay line in the clock phase shifter are transmitted to the digital phase shifter, thereby informing the digital phase shifter of the period of the reference clock signal. In response, the digital phase shifter provides a phase control signal that introduces a delay, which is referenced to the period of the reference clock signal, to either the reference clock signal or the skew clock signal. The phase control signal is proportional to a fraction of the period of the reference clock signal. In one embodiment, the period of the reference clock signal is determined from the tap/trim settings of a delay line in the clock phase shifter. The delay line can have, for example, 512 tap/trim units. The phase control signal is determined by multiplying the equivalent tap/trim units used by a delay line in the clock phase shifter by a fraction. The fraction can be determined by the contents of configuration memory bits stored in an FPGA, or by a user-defined signal.

28 The digital phase shifter can be controlled to operate
29 in one of two fixed modes or in one of two variable modes.
30 In the first fixed mode, the digital phase shifter introduces
31 delay to the skew clock signal. For example, the digital
32 phase shifter can introduce a delay in the range of 0 to 511
33 tap/trim units to the skew clock signal in the first fixed
34 mode. In the second fixed mode, the digital phase shifter
35 introduces delay to the reference clock signal. For example,
36 the digital phase shifter can introduce a delay in the range

1 of 0 to 511 tap/trim units to the reference clock signal in
2 the second fixed mode. In the first variable mode, the
3 digital phase shifter can introduce a delay equal to 255 to -
4 255 tap/trim units to the reference clock signal. In the
5 second variable mode, the digital phase shifter can introduce
6 a delay equal to 255 to -255 tap/trim units to the skew clock
7 signal.

8 In accordance with another embodiment, the digital phase
9 shifter is capable of operating in a low frequency mode or a
10 high frequency mode. The digital phase shifter is controlled
11 to adjust the tap/trim setting provided by the delay line of
12 the clock phase shifter to compensate for different overhead
13 delays experienced by the clock phase shifter in the low
14 frequency mode and the high frequency mode.

15 In yet another embodiment of the present invention, the
16 frequency of the skew clock signal can be dithered around a
17 base frequency, thereby enabling this clock signal to comply
18 with FCC requirements for electromagnetic emissions in many
19 cases. That is, delay can be introduced such that the skew
20 clock signal exhibits slightly different frequencies in
21 successive periods. For example, the frequency of a 100 MHz
22 clock signal can be adjusted to have frequencies of
23 approximately 98, 98.5, 99, 99.5, 100, 100.5, 101, 101.5, and
24 102 MHz during different periods. This configuration is
25 referred to as a "spread-8" configuration, because eight
26 frequencies are generated in addition to the base frequency
27 of 100 MHz. For a 1 MHz window measurement method, because
28 the frequencies are spread in 0.5 MHz increments, only three
29 of the nine frequencies are included in the window. As a
30 result, 2/3 of the energy of the clock signal is not included
31 when determining whether the clock signal meets the FCC
32 electromagnetic emission requirements. By spreading the
33 frequencies above and below the base frequency in a regular
34 manner, the average frequency of the clock signal becomes
35 equal to the base frequency. Other configurations,
36 including, but not limited to, spread-2, spread-4 and spread-

6 configurations, can be implemented in accordance with the present invention.

In a preferred embodiment, the clock frequencies are generated by a digital spread spectrum (DSS) circuit, which operates with the digital phase shifter to insert small delays in the skew clock signal. Because the digital phase shifter delay must be able to adjust both up and down relative to its starting point, the variable mode of the digital phase shifter is typically used in conjunction with the DSS circuit. In accordance with one embodiment, the DSS circuit provides particular patterns of digital tap/trim adjustments to optimize the operation of the digital phase shifter.

In another embodiment, the DSS circuit and/or pattern of digital tap/trim adjustments necessary to successfully implement spread spectrum generation can be used with a conventional delay line, independent of the digital phase shifter.

The present invention will be more fully understood in view of the following description and drawings.

BRIEF DESCRIPTION OF THE DRAWINGS

Figure 1 is a block diagram of a system using a conventional delay lock loop.

Figures 2A, 2B and 2C are timing diagrams for the system of Figure 1.

Figure 3 is a block diagram of a system using an embodiment of a delay lock loop in accordance with the present invention.

Figure 4 is a timing diagram for the delay lock loop of Figure 3.

Figure 5 illustrates a lock window as used in accordance with one embodiment of the present invention.

Figure 6 is a block diagram of an embodiment of a clock phase shifter in accordance with the present invention.

1 Figure 7 is a block diagram of another embodiment of a
2 clock phase shifter in accordance with the present invention.

3 Figure 8 is a block diagram of an output generator in
4 accordance with the present invention.

5 Figure 9 is a state diagram for an embodiment of a
6 controller in accordance with the present invention.

7 Figure 10 is a block diagram of a system using another
8 embodiment of a delay lock loop in accordance with the
9 present invention.

10 Fig. 11 is a block diagram of a delay lock loop, which
11 can be used in place of the delay lock loop of Fig. 3, in
12 accordance with another embodiment of the present invention.

13 Fig. 12 is a schematic diagram illustrating the tap/trim
14 delays for selected sections of a delay line in the clock
15 phase shifter of Fig. 7.

16 Fig. 13 is a block diagram of a digital phase shifter in
17 accordance with one embodiment of the present invention.

18 Figs. 14A and 14B are waveform diagrams illustrating
19 reference clock and skew clock signals for a first fixed mode
20 and a second fixed mode, respectively, of the delay lock loop
21 of Fig. 11.

22 Fig. 14C is a waveform diagram illustrating reference
23 clock and skew clock signals for a first and second variable
24 mode of the delay lock loop of Fig. 11.

25 Fig. 15 is a block diagram illustrating phase shift
26 control logic of Fig. 13 in more detail.

27 Fig. 16 is a block diagram of a digital spread spectrum
28 (DSS) circuit in accordance with another embodiment of the
29 present invention.

30 Fig. 17A is a graph illustrating a DSS bypass mode of
31 the DSS circuit of Fig. 16.

32 Fig. 17B is a graph illustrating a spread spectrum mode
33 of the DSS circuit of Fig. 16.

34 Fig. 18 is a circuit diagram of the DSS circuit of Fig.
35 16 in accordance with one embodiment of the present
36 invention.

Figs. 19A-19B are waveform diagrams illustrating reference clock and skew clock signal that result when the DSS circuit of Fig. 16 is controlled to implement a spread-8 configuration.

DETAILED DESCRIPTION OF THE DRAWINGS

Figure 3 is a block diagram of a system using a delay lock loop 300 in accordance with one embodiment of the present invention. Delay lock loop 300 comprises a delay line 310, a clock phase shifter 350, a controller 330, an output generator 340, and a phase detector 320. Delay lock loop 300 receives reference clock signal REF_CLK on a reference input terminal 302 and generates output clock signal O_CLK on output terminal 304. As explained above with respect to Figure 1, output clock signal O_CLK is skewed by clock skew 180 into skewed clock signal S_CLK, which clocks logic circuits 190. Skewed clock signal S_CLK is also fed back to a feedback terminal 306 of delay lock loop 300 on feedback path 170.

Within delay lock loop 300, reference clock signal REF_CLK is delayed by delay line 310 to generate delayed clock signal D_CLK. Delayed clock signal D_CLK is delayed from clock signal REF_CLK by a propagation delay D in delay line 310. One embodiment of delay lock loop 300 uses an adjustable delay line described in U.S. Patent Application Serial No. 09/102,704 [docket X-440 US], entitled "Glitchless Delay Line Using Gray Code Multiplexer", which is referenced above. However, other adjustable delay lines can also be used with delay lock loop 300. Delayed clock signal D_CLK is provided to an input terminal of a clock phase shifter 350 and to an input terminal of an output generator 340.

Clock phase shifter 350 generates one or more phase-shifted clock signals P_CLK_1 to P_CLK_N-1, where N is a positive integer. In one embodiment, phase-shifted clock signal P_CLK_1 is phase-shifted by $360/N$ degrees from delayed

1 clock signal D_CLK. Phase-shifted clock signal P_CLK_2 is
2 phase-shifted by $2 \cdot (360/N)$ degrees. Phase-shifted clock
3 signal P_CLK_N-1 is phase-shifted by $(N-1) \cdot (360/N)$ degrees.
4 Thus, in general a phase-shifted clock signal P_CLK_Z is
5 phase-shifted by $Z \cdot (360/N)$, where Z is an integer between 1
6 and (N-1), inclusive. Delayed clock signal D_CLK can be
7 considered a phase-shifted clock signal P_CLK_0 since delayed
8 clock signal D_CLK has a 0 degree phase shift from itself.
9 Further, in some embodiments of delay lock loop 300, clock
10 phase shifter 350 generates a phase-shifted signal P_CLK_N
11 that has the same phase and frequency as delayed clock signal
12 D_CLK.

13 Thus, in an embodiment of clock phase shifter 350 where
14 N is equal to four, phase-shifted clock signal P_CLK_1 is
15 phase-shifted 90 degrees from delayed clock signal D_CLK. It
16 logically follows that phase-shifted clock signal P_CLK_2 is
17 phase-shifted by 180 degrees from delayed clock signal D_CLK
18 and phase-shifted clock signal P_CLK_3 is phase-shifted by
19 270 degrees from delayed clock signal D_CLK. However, the
20 principles of the present invention are also suitable for
21 other embodiments of clock phase shifter 350 using other
22 patterns of phase shifting between the phase-shifted clock
23 signals.

24 Phase shifting is a concept in the frequency domain of a
25 clock signal. The equivalent of phase shifting in the time
26 domain is delaying the clock signal. Specifically, if a
27 first clock signal is phase-shifted from a second clock
28 signal by X degrees, the first clock signal is delayed by
29 $X \cdot (P/360)$, where P is the period of the first and second
30 clock signals. Thus, if phase-shifted clock signal P_CLK_1
31 is phase-shifted 90 degrees from delayed clock signal D_CLK,
32 phase-shifted clock signal P_CLK_1 is delayed by one-fourth
33 of the period of delayed clock signal D_CLK. To distinguish
34 delays caused by phase shifting from other propagation
35 delays, delays caused by phase shifting are referred to as
36 phase-shifted delays P_D_Z. Since a phase-shifted clock

1 signal P_CLK_Z is phase-shifted by $Z \cdot (360/N)$ degrees, phase-
2 shifted clock signal P_CLK_Z has a phase-shifted delay P_D_Z
3 equal to $Z \cdot (P/N)$, where Z is an integer between 1 and (N-1),
4 inclusive.

5 Figure 4 illustrates a timing diagram for delay lock
6 loop 300 (Figure 3) wherein N equals 4. Specifically, clock
7 phase shifter 350 generates phase-shifted clock signal
8 P_CLK_1 90 degrees out of phase with delayed clock signal
9 D_CLK. Thus, phase-shifted clock signal P_CLK_1 is delayed
10 by one-fourth of clock period P. Clock phase shifter 350
11 generates phase-shifted clock signal P_CLK_2 180 degrees out
12 of phase with delayed clock signal D_CLK. Thus, phase-
13 shifted clock signal P_CLK_2 is delayed by half of clock
14 period P. Finally, clock phase shifter 350 generates phase-
15 shifted clock signal P_CLK_3 270 degrees out of phase with
16 delayed clock signal D_CLK. Thus, phase-shifted clock signal
17 P_CLK_3 is delayed by three-fourths of clock period P.

18 Returning to Figure 3, clock phase shifter 350 provides
19 the phase-shifted clock signals to various input terminals of
20 output generator 340. In some embodiments of delay lock loop
21 300, clock phase shifter 350 can be configured using one or
22 more configuration signals CFG on an optional configuration
23 bus 360. An embodiment of clock phase shifter 350 that is
24 configured by configuration signals CFG is described below
25 with respect to Figure 7. Configuration signals CFG are
26 received on configuration terminals 308 and are routed to
27 clock phase shifter 350 and controller 330 by configuration
28 bus 360. Output generator 340 selects either delayed clock
29 signal D_CLK or one of the phase-shifted clock signals to
30 provide as output clock signal O_CLK. For embodiments of
31 delay lock loop 300 in which clock phase shifter 350 provides
32 phase-shifted clock signal P_CLK_N, output generator 340 can
33 use phase-shifted clock signal P_CLK_N in place of delayed
34 clock signal D_CLK. Controller 330 controls output generator
35 340.

1 Controller 330 receives phase information regarding
2 reference clock signal REF_CLK and skewed clock signal S_CLK
3 from phase detector 320. Specifically, phase detector 320
4 informs controller 330 whether propagation delay D from delay
5 line 310 should be increased or decreased to achieve
6 synchronization of skewed clock signal S_CLK with reference
7 clock signal REF_CLK. For embodiments of phase detector 320
8 that only determine whether to increase or decrease
9 propagation delay D, a jitter filter (not shown) can be used
10 to reduce clock jitter. In one embodiment, the jitter filter
11 is an up/down counter (not shown) that decrements by one if
12 propagation delay D should be decreased and increments by one
13 if propagation delay D should be increased. However,
14 propagation delay D is not adjusted until the up/down counter
15 reaches 0 or some other predetermined number. When
16 propagation delay D is adjusted, the up/down counter is reset
17 to one-half the maximum value. In other embodiments, phase
18 detector 320 calculates the amount propagation delay D should
19 be increased or decreased. During lock acquisition,
20 controller 330 attempts to synchronize skewed clock signal
21 S_CLK with reference clock signal REF_CLK so that initial
22 propagation delay ID of propagation delay D is within a lock
23 window W.

24 Figure 5 illustrates the concepts of lock window W. As
25 explained above, propagation delay D must be between minimum
26 propagation delay D_MIN and maximum propagation delay D_MAX.
27 Typical values for D_MIN and D_MAX are 3.2 nanoseconds and
28 46.8 nanoseconds, respectively. During lock acquisition,
29 controller 330 ensures that initial propagation delay ID of
30 propagation delay D is within lock window W. Specifically,
31 when synchronization is first established initial propagation
32 delay ID must be between lock window minimum W_MIN and lock
33 window maximum W_MAX. The limits on lock window W are set to
34 guarantee that once delay lock loop 300 completes locks
35 acquisition, delay lock loop 300 can maintain synchronization

1 as long as the system containing delay lock loop 300 operates
2 within the design guidelines of the system.

3 For example, the system containing delay lock loop 300
4 generally can operate in a range of operating conditions.
5 The range of operating conditions includes a maximum extreme
6 condition in which propagation delay SKEW is maximized at a
7 propagation delay value SKEW_MAX. Similarly, the range of
8 operating conditions also includes a minimum extreme
9 condition in which propagation delay SKEW is minimized at a
10 propagation delay value SKEW_MIN. Thus, the maximum change
11 (DELTA_SKEW) in propagation delay SKEW during operation of
12 the system is equal to propagation delay value SKEW_MAX minus
13 propagation delay value SKEW_MIN (i.e., $\text{DELTA_SKEW} = \text{SKEW_MAX}$
14 $- \text{SKEW_MIN}$). For maximum protection during lock maintenance,
15 lock window minimum W_MIN can be equal to minimum propagation
16 delay D_MIN plus DELTA_SKEW. Similarly, lock window maximum
17 W_MAX can be equal to maximum propagation delay D_MAX minus
18 DELTA_SKEW. In one embodiment of the present invention, lock
19 window minimum W_MIN is equal to approximately 16.5% of
20 maximum propagation delay D_MAX and lock window maximum W_MAX
21 is equal to approximately 67.8% of maximum propagation delay
22 D_MAX.

23 As explained above with respect to Figure 1, for a
24 conventional delay lock loop synchronization of skewed clock
25 signal S_CLK with reference clock signal REF_CLK is achieved
26 when propagation delay D plus propagation delay SKEW is equal
27 to a multiple of period P. In equation form:

$$28 \quad D + \text{SKEW} = \text{MULT}(P) \quad (1)$$

29
30 where MULT(P) refers to a multiple of P. Usually, the
31 smallest multiple of P greater than SKEW is used.

32 With delay lock loop 300, controller 330 can also use
33 the delays from the phase-shifted clock signals. Thus delay
34 lock loop 300 can achieve synchronization if propagation
35 delay D plus a phase-shifted delay P_D from a phase-shifted
36

1 clock signal plus propagation delay SKEW is a multiple of
2 period P. In equation form:

$$D + P_D_Z + SKEW = MULT(P) \quad (2)$$

3
4
5
6 where P_D_Z refers to a phase-shifted delay from phase-
7 shifted clock signal P_CLK_Z. Usually, the smallest multiple
8 of P greater than propagation delay SKEW plus phase-shifted
9 delay P_D_Z is used. As explained above with respect to
10 Figure 3, in one embodiment of clock phase shifter 350 phase-
11 shifted delay P_D_Z of a phase-shifted clock signal P_CLK_Z
12 is equal to $Z*(P/N)$, where Z is an integer between 0 and (N-
13 1), inclusive. If Z is equal to 0, controller 330 causes
14 output generator 340 to use delayed clock signal D_CLK as
15 output clock signal O_CLK. Thus, phase-shifted delay P_D_0
16 is equal to 0.

17 For clarity, initial delay ID can be referred to initial
18 delay ID_0 if output generator 340 uses delayed clock signal
19 D_CLK for output clock signal O_CLK. Similarly, initial
20 delay ID can be referred to as initial delay ID_Z, if output
21 generator 340 uses phase-shifted clock signal P_CLK_Z for
22 output clock signal O_CLK, where Z is a positive integer
23 between 1 and (N-1), inclusive. Thus, at the end of lock
24 acquisition, equation (2) can be rewritten as:

$$ID_Z + P_D_Z + SKEW = MULT(P) \quad (3)$$

25
26
27
28 Re-arranging equation (3) provides:

$$ID_Z = MULT(P) - SKEW - P_D_Z \quad (4)$$

29
30
31
32 and substituting $Z*(P/N)$ for P_D_Z provides:

$$ID_Z = MULT(P) - SKEW - Z*(P/N) \quad (5)$$

1 Usually, the smallest multiple of P that results in a
2 positive initial delay ID_Z is used. In situations where
3 initial delay ID_Z is less than minimum propagation delay
4 D_MIN or greater than maximum propagation delay D_MAX, delay
5 lock loop 300 cannot synchronize skewed clock signal S_CLK
6 with reference clock signal REF_CLK using phase-shifted clock
7 signal P_CLK_Z.

8 Because controller 330 can select any one of phase-
9 shifted clock signals P_CLK_Z to drive output clock signal
10 O_CLK, controller 330 can select from N initial delay values.
11 The possible initial delay values range from a minimum offset
12 value ($MULT(P) - SKEW$) to a maximum value ($MULT(P) - SKEW$
13 $+ (N-1)/(N * \text{period } P)$). The difference between each
14 initial delay value is period P divided by N. For example,
15 if N equals four, period P equals 40 nanoseconds, and
16 propagation delay SKEW equals 25 nanoseconds; then initial
17 delays ID_0, ID_1, ID_2, and ID_3 equal 15 nanoseconds, 5
18 nanoseconds, 35 nanoseconds, and 25 nanoseconds, respectively
19 (as calculated using equation (5)). If N equals four, period
20 P equals 40 nanoseconds, and propagation delay SKEW equals 55
21 nanoseconds; then initial delays ID_0, ID_1, ID_2, and ID_3
22 equal 25 nanoseconds, 15 nanoseconds, 5 nanoseconds, and 35
23 nanoseconds, respectively. Thus, controller 330 is likely to
24 find one or more initial delay values within lock window W.
25 If more than one initial delay value is within lock window W,
26 controller 330 can select any one of the initial delay values
27 within lock window W.

28 Some embodiments of controller 330 can perform the
29 calculations described above to determine which phase-shifted
30 clock signal P_CLK_Z to use. However, other embodiments use
31 trial and error to determine which phase-shifted clock signal
32 P_CLK_Z to use. An embodiment of controller 330 that uses
33 trial and error is described below with respect to Figure 9.

34 Figure 6 illustrates one embodiment of clock phase
35 shifter 350 of Figure 3. The embodiment of clock phase
36 shifter 350 in Figure 6 comprises a phase detector 620 and a

1 plurality of delay lines 610_1 to 610_N. Delay lines 610_1
2 to 610_N are coupled in series. The input terminal of delay
3 line 610_1 receives an input clock signal such as delayed
4 clock signal D_CLK (Figure 3). The output terminal of delay
5 line 610_N is coupled to an input terminal of phase detector
6 620. Phase detector 620 also receives input clock signal
7 D_CLK on another input terminal. Phase detector 620 controls
8 all the delay lines in parallel via control line 625, and
9 each delay line provides the same amount of propagation
10 delay. Consequently, input clock signal D_CLK and the clock
11 signal P_CLK-N on the output terminal of delay line 610_N are
12 synchronized, i.e., in phase. Further, phase detector 620
13 causes the total propagation delay generated by delay lines
14 610_1 to 610_N to be equal to one period P of the input
15 clock. Thus, each delay line provides a propagation delay of
16 P/N . Thus, the output terminal of delay line 610_1 provides
17 a clock signal that is delayed from the input clock signal by
18 P/N whereas the output terminal of delay line 610_2 provides
19 a clock signal that is delayed from the input clock signal by
20 $2*P/N$. In general, the output terminal of delay line 610_Z
21 provides a clock signal that is delayed from the input clock
22 signal by $Z*P/N$, where Z is an integer between 1 and N-1,
23 inclusive. Accordingly, if the input clock signal is delayed
24 clock signal D_CLK, the output terminals of delay lines 610_1
25 to 610_N-1 provide phase-shifted clock signals P_CLK_1 to
26 P_CLK_N-1, respectively. Some embodiments of clock phase
27 shifter 350 also generate a clock signal P_CLK_N on the
28 output terminal of delay line 610_N that has the same phase
29 as delayed clock signal D_CLK.

30 Figure 7 shows a configurable embodiment of clock phase
31 shifter 350 of Figure 3. Specifically, the clock phase
32 shifter of Figure 7 can be configured in a first mode to
33 produce three phase-shifted clock signals that are 90
34 degrees, 180 degrees, and 270 degrees out of phase with an
35 input clock signal. In a second mode, the clock phase
36 shifter of Figure 7 produces a single phase-shifted clock

signal that is 180 degrees out of phase with the input clock signal. The clock phase shifter of Figure 7 comprises a phase detector 720, delay lines 710_1, 710_2, 710_3, and 710_4, and multiplexers 730_1, 730_2, 730_3, and 730_4. A configuration line 740 is coupled to the select terminal of multiplexers 730_1 to 730_4.

The input terminal of delay line 710_1 is coupled to receive an input clock signal such as delayed clock signal D_CLK (Figure 3). The output terminal of each delay line 710_Z is coupled to the logic one input terminal of multiplexer 730_Z, where Z is an integer between 1 and 3, inclusive. The output terminal of each multiplexer 730_Z is coupled to the input terminal of delay line 710_Z+1, where Z is an integer between 1 and 3, inclusive. The output terminal of multiplexer 730_4 is coupled to an input terminal of phase detector 720. The logic zero input terminals of multiplexer 730_1 and multiplexer 730_3 are coupled to ground. However, the logic zero input terminal of multiplexer 730_2 is coupled to the output terminal of delay line 710_1. Similarly, the logic zero input terminal of multiplexer 730_4 is coupled to the output terminal of delay line 710_3. Phase detector 720 also receives input clock signal D_CLK on another input terminal. Phase detector 720 controls delay lines 710_1 to 710_4 in parallel as described above with respect to phase detector 620.

If configuration line 740 is pulled to logic one, which puts the embodiment of Figure 7 into the first mode, delay lines 710_1 to 710_4 are coupled in series. In the first mode, each delay line provides a delay of $P/4$. Thus, if the input clock signal is delayed clock signal D_CLK, the output terminal of each multiplexer 730_Z can provide phase-shifted clock signals P_CLK_1, P_CLK_2, and P_CLK_3.

However, if configuration line 740 is pulled to logic zero, which puts the embodiment of Figure 7 into the second mode, only delay lines 710_1 and 710_3 are coupled in series. Delay lines 710_2 and 710_4 have their input terminal coupled

1 to ground through multiplexers 730_1 and 730_3, respectively.
2 In the second mode delay lines 710_1 and 710_3 each provide a
3 delay of $P/2$. Coupling the input terminals of delay lines
4 710_2 and 710_4 to ground reduces power consumption and
5 switching noise. However, in the second mode the embodiment
6 of Figure 7 produces only one output clock signal, which is
7 180 degrees out of phase with the input clock signal and is
8 generated at the output terminal of multiplexer 730_2.

Figure 8 shows one embodiment of output generator 340 of Figure 3. The output generator of Figure 8 comprises an N-input multiplexer 810. N-input multiplexer 810 has N input terminals, referenced as 810_0 to 810_N-1, select terminals 812, and an output terminal 814. When the embodiment of output generator 340 of Figure 8 is used in delay lock loop 300 of Figure 3, select terminals 812 are coupled to controller 330, input terminal 810_0 is coupled to receive delayed clock signal D_CLK, output terminal 814 provides output clock signal O_CLK, and input terminals 810_1 to 810_N-1 are coupled to receive phase-shifted clock signals P_CLK_1 to P_CLK_N-1, respectively. Select signals on select terminals 812 determine which input signal is provided on output terminal 814. Other embodiments of output generator 340 may include additional circuitry, such as clock buffers and clock dividers. In addition, some embodiments of output generator 340 drive additional clock signals, such as various versions of the phase-shifted clock signals.

Figure 9 shows a state diagram 900 for one embodiment of controller 330 of Figure 3. On power-up or reset, controller 330 transitions to reset stage 910. In reset stage 910, controller 330 sets a phase counter (not shown) to zero, which causes output generator 340 to provide delayed clock signal D_CLK as output clock signal O_CLK, and adjusts propagation delay D of delay line 310 (Figure 3) to a starting delay value. Starting delay values for propagation delay D include, for example, minimum propagation delay D MIN, maximum propagation delay D_MAX, or the average of

1 minimum propagation delay D_MIN and maximum propagation delay
2 D_MAX. Controller 910 then transitions to lock acquisition
3 stage 920.

4 In lock acquisition stage 920, controller 330
5 synchronizes reference clock signal REF_CLK and skewed clock
6 signal S_CLK. Specifically, controller 330 adjusts
7 propagation delay D of delay line 310 based on signals from
8 phase detector 320. Phase detector 320 determines whether
9 propagation delay D must be increased or decreased to
10 synchronize skewed clock signal S_CLK with reference clock
11 signal REF_CLK. Lock acquisition is described above in
12 greater detail with respect to Figures 3-6; therefore, the
13 description is not repeated. In some embodiments, clock
14 phase shifter 350 is also reset by the power-on/reset signal.
15 For some of these embodiments, controller 330 does not adjust
16 propagation delay D until after clock phase shifter 350
17 produces phase-shifted clock signals P_CLK_1 to P_CLK_N-1.
18 If controller 330 cannot synchronize skewed clock signal
19 S_CLK with reference clock signal REF_CLK, controller 330
20 transitions to increment phase stage 950, described below.
21 Otherwise, controller 330 transitions to check lock window
22 stage 930 after controller 330 synchronizes skewed clock
23 signal S_CLK with reference clock signal REF_CLK (with an
24 initial propagation delay ID in delay line 310).

25 In check lock window stage 930, controller 330 must
26 determine whether initial propagation delay ID is within lock
27 window W. Specifically, propagation delay ID is within lock
28 window W if propagation delay ID is greater than lock window
29 minimum W_MIN and less than lock window maximum W_MAX. If
30 initial propagation delay ID is not within lock window W,
31 controller 330 transitions to increment phase stage 950.
32 Otherwise, controller 330 transitions to lock maintenance
33 stage 940.

34 In lock maintenance stage 940, controller 330 adjust
35 propagation delay D of delay line 310 to maintain
36 synchronization of skewed clock signal S_CLK with reference

1 clock signal REF_CLK. Lock maintenance is described above in
2 greater detail; therefore, the description is not repeated.
3 As described above, the present invention can maintain lock
4 throughout the systems environment conditions. Therefore,
5 controller 330 remains in lock maintenance stage 940 unless a
6 reset occurs that causes controller 330 to transition to
7 reset stage 910.

8 In increment phase stage 950, controller 330 increments
9 the phase counter, which causes output generator 340 to
10 select a different phase-shifted clock signal. Further,
11 controller 330 resets delay line 310 so that propagation
12 delay D returns to the starting delay value used in reset
13 stage 910. Controller 330 then transitions to lock
14 acquisition stage 920 and proceeds as described above.

15 Figure 10 is a block diagram of another embodiment of
16 delay lock loop 300. The embodiment of Figure 10 uses the
17 same principles as described above with respect to the
18 embodiment of Figure 3. However, in the embodiment of Figure
19 10, clock phase shifter 350 generates phase-shifted clock
20 signals P_CLK_1 to P_CLK_N-1 using reference clock signal
21 REF_CLK. Reference clock signal REF_CLK and phase-shifted
22 clock signals P_CLK_1 to P_CLK_N-1 are coupled to an input
23 selector 1040. Input selector 1040 selects either reference
24 clock signal REF_CLK or one of phase-shifted clock signals
25 P_CLK_1 to P_CLK_N-1 as a delay line input clock signal
26 DLI_CLK, which is provided to the input terminal of delay
27 line 310. Delay line 310 drives output clock signal O_CLK.
28 A controller 1030 controls input selector 1040 and delay line
29 310 based on the phase information provided by phase detector
30 320 so that delay line 310 provides a propagation delay D
31 that synchronizes skewed clock signal S_CLK with reference
32 clock signal REF_CLK. Input selector 1040 can be implemented
33 using the same circuit design as output generator 340.

34 In the various embodiments of the present invention,
35 novel structures have been described for delay lock loops.
36 By using a clock phase shifter to provide propagation delays

THE BIBLE

25

1 shifted feedback clock signal PS_S_CLK is provided to an
2 input terminal of phase detector 320. Thus, the PS_S_CLK
3 signal of delay lock loop 400 is routed in the same manner as
4 the skew clock signal S_CLK of delay lock loop 300.

5 As described in more detail below, digital phase shifter
6 1100 adjusts the phase relationship of the REF_CLK and S_CLK
7 signals to provide the PS_REF_CLK and PS_S_CLK signals,
8 respectively. As a result, the S_CLK signal can be
9 controlled to have a leading or lagging phase relationship
10 with respect to the REF_CLK signal.

11 In the described embodiment, clock phase shifter 350 is
12 configured in the manner illustrated in Fig. 7 (i.e., $N=4$).
13 Each of the delay lines 710_1, 710_2, 710_3 and 710_4
14 includes 128 tap delays and a trim delay circuit. The trim
15 delay circuit can be controlled to add up to 3 trim delays
16 between tap delays. Delay line 710_3 therefore has 512 (128
17 $+ 3 \cdot 128$) possible tap/trim delay settings. Fig. 12
18 illustrates the tap/trim delays for selected sections of
19 delay line 710_3. Circuitry for providing the tap/trim
20 delays is described in commonly owned, co-pending U.S. Patent
21 Application Serial No. 09/102,704, entitled "Glitchless Delay
22 Line Using Gray Code Multiplexer" by Andrew K. Percey, which
23 is incorporated herein by reference.

24 Phase detector 720 controls each of the delay lines
25 710_1, 710_2, 710_3 and 710_4 to have the same tap/trim
26 setting (± 1 trim delay). When the configuration signal
27 CFG has a logic "1" value (i.e., delay lock loop 400 is
28 configured in a low frequency mode), all four of the delay
29 lines 710_1, 710_2, 710_3 and 710_4 are coupled in series.
30 As a result, the delay selected by the tap/trim setting of
31 each of the delay lines corresponds with approximately one-
32 quarter cycle of the D_CLK signal.

33 Similarly, when the configuration signal CFG has a logic
34 "0" value (i.e., delay lock loop 400 is configured in a high
35 frequency mode), the two delay lines 710_1 and 710_3 are
36 coupled in series. As a result, the delay selected by the

4 Clock phase shifter 350 provides the tap/trim setting of
5 delay line 710_3 to digital phase shifter 1100, thereby
6 providing digital phase shifter 1100 with a signal that
7 corresponds with the period of the D_CLK signal. As
8 described in more detail below, this information is used to
9 select the phase shift introduced by digital phase shifter
10 1100.

18 Digital phase shifter 1100 is controlled as follows.
19 First, digital phase shifter 1100 is selected to operate in
20 one of four modes. These four modes include a first fixed
21 mode, a second fixed mode, a first variable mode and a second
22 variable mode. In the first fixed mode, digital phase
23 shifter 1100 introduces delay to the skew clock signal. For
24 example, digital phase shifter 1100 can be controlled to
25 introduce a delay in the range of 0 to 511 tap/trim units to
26 the skew clock signal in the first fixed mode. In the second
27 fixed mode, digital phase shifter 1100 introduces delay to
28 the reference clock signal. For example, digital phase
29 shifter 1100 can be controlled to introduce a delay in the
30 range of 0 to 511 tap/trim units to the reference clock
31 signal in the second fixed mode. In the first variable mode,
32 digital phase shifter 1100 can be controlled to introduce a
33 delay equal to 255 to -255 tap/trim units to the reference
34 clock signal. In the second variable mode, digital phase
35 shifter 1100 can be controlled to introduce a delay equal to
36 255 to -255 tap/trim units to the skew clock signal.

1 The mode is selected in response to the S_LAGS_REF and
2 CENTERED control signals. The CENTERED control signal is de-
3 asserted low, and the S_LAGS_REF signal is de-asserted low to
4 indicate that the S_CLK signal will lead the REF_CLK signal
5 in the first fixed mode. Conversely, the CENTERED control
6 signal is de-asserted low, and the S_LAGS_REF signal is
7 asserted high to indicate that the S_CLK signal will lag the
8 REF_CLK signal in the second fixed mode. The CENTERED
9 control signal is asserted high and the S_LAGS_REF signal is
10 asserted high to enable the first variable mode. The
11 CENTERED control signal is asserted high and the S_LAGS_REF
12 signal is de-asserted low to enable the second variable mode.
13 In one embodiment, these control signals are provided by
14 configuration memory bits of a programmable logic device,
15 although this is not necessary.

16 Fig. 14A is a waveform diagram illustrating the REF_CLK
17 and S_CLK signals for the first fixed mode. To enter the
18 first fixed mode, the S_LAGS_REF control signal is de-
19 asserted to a logic "0" state, and the CENTERED control
20 signal is de-asserted to a logic "0" state. Under these
21 conditions, the REF_CLK signal is routed through multiplexers
22 M0, M2 and M6 and overhead delay circuit 1301 to provide the
23 PS_REF_CLK signal. The S_CLK signal is routed through
24 multiplexers M1, M4 and M7, overhead delay circuit 1302 and
25 512-tap/trim delay line 1304 to provide the PS_S_CLK signal.
26 Overhead delay circuits 1301 and 1302 introduce the same
27 delay to the applied signals. Thus, if the 512-tap/trim
28 delay line 1304 is set to have zero delay, then the REF_CLK
29 signal and the S_CLK signal will have identical delays
30 through digital phase shifter 1100.

31 Delay lock loop 400 will always cause the PS_REF_CLK and
32 the PS_S_CLK signals to be synchronized. As a result, any
33 phase shifting introduced by delay elements 1301-1304 is
34 realized by the REF_CLK and S_CLK signals. In the first
35 fixed mode, if the delay introduced by 512-tap/trim delay
36 line 1304 is increased, then the REF_CLK signal will lag the

1 S_CLK signal. The 512-tap/trim delay line 1304, which is
2 identical to delay line 710_3 (Fig. 7), can be configured to
3 provide a maximum tap/trim delay of 511 trim units (thereby
4 providing a maximum lag for the REF_CLK signal).

5 Fig. 14B is a waveform diagram illustrating the REF_CLK
6 and S_CLK signals for the second fixed mode. To enter the
7 second fixed mode, the S_LAGS_REF control signal is asserted
8 to a logic "1" state, and the CENTERED control signal is de-
9 asserted to a logic "0" state. Under these conditions, the
10 REF_CLK signal is routed through multiplexers M1, M2 and M6,
11 overhead delay circuit 1302 and 512-tap/trim delay line 1304
12 to provide the PS_REF_CLK signal. The S_CLK signal is routed
13 through multiplexers M0, M4 and M7 and overhead delay circuit
14 1301 to provide the PS_S_CLK signal. If the 512-tap/trim
15 delay line 1304 is set to have zero delay, then the REF_CLK
16 signal and the S_CLK signal will have identical delays
17 through digital phase shifter 1100.

18 In the second fixed mode, if the delay introduced by
19 512-tap/trim delay line 1304 is increased, then the REF_CLK
20 signal will begin to lead the S_CLK signal. When the 512-
21 tap/trim delay line 1304 is configured to provide a maximum
22 tap/trim delay of 511 trim units, a maximum phase lead will
23 be provided for the REF_CLK signal.

24 Fig. 14C is a waveform diagram illustrating the REF_CLK
25 and S_CLK signals for the first and second variable modes.
26 To enter the first variable mode, the CENTERED control signal
27 is asserted to a logic "1" state and the S_LAGS_REF control
28 signal is asserted to a logic "1" state. As a result, the
29 REF_CLK signal is routed through multiplexers M1, M3 and M6,
30 overhead delay circuit 1302 and 512-tap/trim delay line 1304
31 to provide the PS_REF_CLK signal. The S_CLK signal is routed
32 through multiplexers M0, M5 and M7, overhead delay circuit
33 1302 and 64-tap delay line 1303. The 64-tap delay circuit
34 1303 provides a fixed delay equal to half of the maximum
35 delay of 512-tap/trim delay line 1304 (or delay line 710_3).
36 The 512-tap/trim delay line 1304 is initially set to a delay

Fig. 14C also represents the REF_CLK and S_CLK signals for the second variable mode. To enter the second variable mode, the CENTERED control signal is asserted to a logic "1" state and the S_LAGS_REF control signal is asserted to a logic "0" state. As a result, the REF_CLK signal is routed through multiplexers M0, M3 and M4, overhead delay circuit 1301 and 64-tap delay line 1303 to provide the PS_REF_CLK signal. The S_CLK signal is routed through multiplexers M1, M6 and M7, overhead delay circuit 1301 and 512-tap/trim delay line 1304. The 512-tap/trim delay line 1304 is initially set to a delay equal to zero trim units, and is incremented to a delay equal to 64-taps (256 trim units). At this time, the delays through delay circuits 1303 and 1304 are initially matched (i.e., there is a zero phase shift). The delay introduced by the 512-tap/trim delay line 1304 can be increased greater than the 64 tap delay, thereby causing the S_CLK signal to lead the REF_CLK signal. Conversely, the delay introduced by the 512-tap/trim delay line 1304 can be reduced to less than the 64 tap delay, thereby causing the S_CLK signal to lag the REF_CLK signal. The second variable mode is similar to the first variable mode. However, in the first variable mode, 512-tap/trim delay line 1304 is in line with the REF_CLK signal, and in the second variable mode, the

1 512-tap/trim delay line 1304 is in line with the S_CLK
2 signal. In the described embodiments, the first variable
3 mode is preferred for the following reason. If the 512-
4 tap/trim delay line 1304 is in line with the S_CLK signal,
5 then delay line 310 will subsequently need to compensate for
6 adjustments made by delay line 1304, thereby reducing the
7 number of available tap adjustments remaining in delay line
8 310.

9 Delay lock loop 400 operates as follows in accordance
10 with one embodiment of the present invention.

11 Initially, configuration memory bits are set to define
12 the states of the S_LAGS_REF and CENTERED control signals,
13 thereby defining whether digital phase shifter 1100 will
14 operate in the first fixed mode (REF_CLK lagging S_CLK), the
15 second fixed mode (REF_CLK leading S_CLK) or the first or
16 second variable mode.

17 Configuration memory bits are also set to define the
18 states of the PS_SIGN, PS_MAG[7:0], HF_MODE, LFC and HFC
19 control signals. The PS_MAG[7:0] control signal identifies
20 the magnitude of the phase shift to be introduced by digital
21 phase shifter 1100. The PS_SIGN control signal is used to
22 identify the polarity of the PS_MAG[7:0] signal when digital
23 phase shifter 1100 is configured in the variable mode (i.e.,
24 CENTERED = 1). When digital phase shifter 1100 is configured
25 in a fixed mode, the PS_SIGN control signal is not used. If
26 the PS_SIGN signal has a logic "1" state (indicating a
27 negative polarity), the PS_MAG[7:0] signal cannot have a
28 logic 0 value, because the PS_MAG[7:0] signal is represented
29 in 2's complement in the present embodiment.

30 The HF_MODE control signal is set to a logic "0" value
31 when digital phase shifter 100 is operating in response to
32 low frequency clock signals, where all four delay lines
33 710_1-710_4 are required to create one full period.
34 Conversely, the HF_MODE control signal is set to a logic "1"
35 value when digital phase shifter 1100 is operating in
36 response to high frequency clock signals, where only two

The low frequency mode constant LFC[7:0] represents the overhead of the signal path in trim units through the delay chain located in clock phase shifter 350 (Fig. 7), when the CFG signal has a logic "1" value and all of the tap/trim settings in delay lines 710_1-710_4 are set to zero.

After the above-described constants have been set, the RESET signal is asserted, thereby resetting up/down counter 1311 to a value corresponding with zero tap/trim delay in delay line 1304. Delay line 310 is also set to a value corresponding with zero tap/trim delay. These settings are maintained until after clock phase shifter 350 achieves a locked condition. More specifically, the REF_CLK signal is routed through digital phase shifter 1100 and delay line 310, and is provided to clock phase shifter 350 (as the D_CLK signal). In response, clock phase shifter 350 operates in the manner described above in connection with Fig. 7 to achieve a locked condition with respect to the REF_CLK signal. Note that digital phase shifter 1100 and delay line 310 are prevented from adjusting their delay lines while clock phase shifter 350 is locking using the REF_CLK signal. After clock phase shifter 350 has achieved a locked condition, the tap/trim setting of delay line 710_3 is representative of either $\frac{1}{4}$ of the period (low frequency mode) or $\frac{1}{2}$ of the period (high frequency mode) of the reference

1 clock signal REF_CLK. At the end of this state, the tap/trim
2 setting of delay line 710_3 is provided to phase shift
3 control logic 1310 as the TAP_TRIM[8:0] signal. The RESET
4 signal is de-asserted after the clock phase shifter 350 is
5 locked.

6 After the RESET signal is de-asserted, digital phase
7 shifter 1100 is allowed to adjust delay line 1304. However,
8 while digital phase shifter 1100 is adjusting delay line
9 1304, clock phase shifter 350 and delay line 310 are
10 prevented from adjusting their delay lines. Digital phase
11 shifter 1100 increments counter 1311 to provide the initial
12 setting of 512-tap/trim delay line 1304. The initial setting
13 of 512-tap/trim delay line 1304 is calculated as a function
14 of the TAP_TRIM[8:0] signal and the PS_SIGN and PS_MAG[6:0]
15 values. Up/down counter 1311 is incremented (or decremented)
16 until the count of this counter 1311 matches the calculated
17 initial setting. This count is transparently passed to 512-
18 tap/trim delay line 1304 through binary-to-gray code
19 converter 1305. Binary-to-gray code converter is described
20 in more detail in commonly owned, co-pending U.S. Patent
21 Application Serial No. 09/102,704.

22 More specifically, the initial setting of 512-tap/trim
23 delay line 1304 is determined by first determining an
24 equivalent tap/trim per period (ETT/P). For low frequency
25 mode, ETT/P is determined by the following equation.

26
27
$$\text{ETT/P} = (4 \times (\text{TAP_TRIM}[8:0])) + \text{LFC}[7:0] \quad (6)$$

28

29 For high frequency mode, ETT/P is determined by the
30 following equation.

31
32
$$\text{ETT/P} = (2 \times (\text{TAP_TRIM}[8:0])) + \text{HFC}[7:0] \quad (7)$$

33

34 The initial setting of 512-tap/trim delay line 1304 is
35 then determined from the ETT/P value in the following manner.
36 For the first and second fixed modes of operation, the

1 initial tap/trim setting of 512-tap/trim delay line 1304 is
2 equal to:

3
4
$$(PS_MAG[7:0]/256) \times ETT/P \quad (8)$$

5
6 For the first and second variable modes of operation,
7 the initial tap/trim setting of 512-tap/trim delay line 1304
8 is equal to:

9
10
$$256 + ((PS_MAG[7:0]/256) \times ETT/P); \quad (9)$$

11 If $PS_SIGN = 0$

12
13 or

14
15
$$256 - ((PS_MAG[7:0]/256) \times ETT/P); \quad (10)$$

16 If $PS_SIGN = 1$.

17
18 Because there is only one delay line 1304 in digital
19 phase shifter 1100, the maximum value of $PS_MAG[7:0]/256$ is
20 less than approximately $\frac{1}{4}$ and $\frac{1}{2}$ of the longest period, for
21 the low and high frequency modes, respectively.

22 After the initial tap/trim setting of 512-tap/trim delay
23 line 1304 has been set, digital phase shifter 1100 and clock
24 phase shifter 350 are temporarily prevented from adjusting
25 their delay lines. At this time, delay line 310 is released,
26 thereby enabling delay line 310 to be adjusted, such that the
27 PS_REF_CLK and the PS_S_CLK signals are synchronized. That
28 is, delay line 310 is allowed to achieve a locked condition.
29 At this time, the S_CLK and REF_CLK signals exhibit a skew
30 corresponding with the delay introduced by 512-tap/trim delay
31 line 1304. In general processing continues in the above-
32 described manner, such that during the next state, clock
33 phase shifter 350 is allowed to lock, while delay line 1304
34 in digital phase shifter 1100 and delay line 310 are held at
35 their previously determined values. During the next state,
36 delay line 1304 of digital phase shifter 1100 is allowed to

The tap/trim setting of delay line 1304 can also be modified by the user of delay lock loop 400 through a user interface 1320. User interface 1320 includes the phase increment/decrement signal PSINCDEC, the phase shift enable signal PSEN, the phase shift clock signal PSCLK and the phase shift done signal PSDONE, which are provided to phase shift control logic 1310. The PSCLK signal is different than the PS_DLY_OUT clock signal, thereby requiring the coordination of these two clock domains within phase shift control logic 1310. The PSEN signal is asserted high for one cycle of the

1 PSCLK signal. At the same time, or prior to this cycle, the
2 PSINCDEC signal is asserted high or low, thereby causing the
3 numerator of the fraction in equations (8), (9) or (10) to be
4 incremented or decremented by one, respectively. If the
5 increment or decrement in the numerator is sufficient to
6 warrant a change in the count value in up/down counter 1311,
7 then this change is implemented. When digital phase shifter
8 1100 has completed the increment or decrement operation,
9 phase shift control logic 1310 asserts the PSDONE signal for
10 one cycle of the PSCLK signal, thereby indicating to the user
11 that the tap/trim setting of delay line 1304 can be modified
12 again.

Fig. 15 is a block diagram illustrating phase shift control logic 1310 in more detail. In the described embodiment, phase shift control logic 1310 includes multiplexers 1501-1506, up/down signed counter 1511, registers M0-M1, adder 1531, bias adder/subtractor 1532, comparator block 1533, product register 1541, OR gate 1542, AND gate 1543, control block 1550 and up/down counter 1311.

Phase shift control logic 1310 performs the mathematics to convert the TAP_TRIM[8:0], LFC[7:0], HFC[7:0], PS_SIGN and PS_MAG[7:0] signals to the PS_TT[8:0] value, such that the phase shift delay introduced by delay line 1304 is the desired fractional part of the clock period. Phase shift control logic 1310 also includes a user interface 1320 that allows dynamic phase adjustments, and an interface to DLL control 1312.

```

28         DLL control 1312 specifies when phase shift control
29         logic 1310 may change the PS_TT[8:0] signal.  DLL control
30         1312 does this with an asynchronous 4 wave front hand shake
31         cycle between the GO signal (a request from DLL control 1312)
32         and the DONE signal (a response from phase shift control
33         logic 1310).

```

34 The first GO signal asserted after the RESET signal
35 causes the PS_TT[8:0] signal to increment from zero to the
36 correct initial setting. On subsequent cycles, the

1 LFC[7:0] constant) is loaded into product register 1541,
2 thereby completing the M1_TO_PROD state.

3 The primary state machine then enters a TAP_TRIM_TO_M1
4 state, where the TAP_TRIM[8:0] signal is loaded into register
5 M1. The primary state machine controls the multiplexers
6 1502-1504 to route the TAP_TRIM[8:0] signal to register M1.

7 The primary state machine then enters a SHIFT1 **state**, in
8 which the TAP_TRIM[8:0] signal is multiplied by 2. This is
9 accomplished by routing the TAP_TRIM[8:0] signal from the
10 output of register M1 to the "1" input terminal of
11 multiplexer 1506. This path shifts the TAP_TRIM[8:0] signal
12 left by one bit, thereby effectively multiplying the
13 TAP_TRIM[8:0] signal by 2. This corresponds with the
14 multiply by 2 function described above in equation (7). The
15 shifted TAP_TRIM[8:0] signal is routed through multiplexers
16 1506, 1503 and 1504 and is re-loaded into register M1.

17 If the HF_MODE signal has a logic "0" state (i.e., low
18 frequency mode), then the primary state machine enters a
19 SHIFT2 state, in which the TAP_TRIM[8:0] signal is again
20 multiplied by 2 (thereby providing the multiply by four
21 function required by equation (6)). The multiply by 2
22 operation of the SHIFT2 state is performed in the same manner
23 as the multiply by 2 operation of the SHIFT1 state. If the
24 HF_MODE signal has a logic "1" state (i.e., high frequency
25 mode), then the primary state machine skips the SHIFT2 state.

26 The primary state machine then enters a ETPP_TO_M1
27 state, in which the shifted TAP_TRIM[8:0] signal stored in
28 register M1 is added to the LFC[7:0] constant stored in
29 product register 1541, thereby creating the ETT/P value. To
30 accomplish this, adder 1531 is controlled to add the contents
31 of register M1 (i.e., the shifted TAP_TRIM[8:0] value) and
32 the contents of product register 1541 (i.e., the LFC[7:0]
33 constant). The result (i.e., the ETT/P value) is routed
34 through multiplexer 1504 and loaded into register M1.

0954523-100600

34 This process is repeated, with the ETT/P value stored in
35 register M1 being shifted left and the PS[7:0] signal in
36 register M0 being shifted right on each iteration. The

After the MULTIPLY state is complete, the primary state machine enters a PASS state. During the PASS state, bias adder/subtractor 1532 is allowed time to complete an add or subtract operation. Bits P[16:8] of product register 1541 are provided to the "A" input terminal of bias adder/subtractor 1532, thereby right shifting the contents of product register 1541 by 8 bits. This effectively divides the contents of product register by 256 (see equations (8), (9), (10)). The "B" input terminal of bias adder/subtractor 1532 is coupled to receive a value of either "0" (if CENTERED = 0) or 256 (if CENTERED = 1). Bias adder/subtractor 1532 will perform an addition (B+A) if AND gate 1543 provides a logic "0" output signal. Conversely, bias adder/subtractor 1532 will perform a subtraction (B-A) if AND gate 1543 provides a logic "1" output signal. AND gate 1543 only provides a logic "1" signal if delay lock loop 400 is configured in the variable mode (CENTERED = 1), and up/down signed counter 1511 has a negative value (SIGN = 1). If bit R[9] of bias adder/subtractor 1532 has a logic "1" value, OR gate 1542 will assert the OVERFLOW signal, thereby indicating that an overflow condition exists. As long as the OVERFLOW signal is not asserted, the result R[8:0] provided by bias adder/subtractor 1532 represents the desired PS_TT[8:0] signal.

1 1FF, inclusive (c) tap/trim counter 1311 would wrap if
2 changed in the selected direction, and (d) up/down counter
3 1511 would wrap if changed in the selected direction.

4 After the PASS state is complete, the primary state
5 machine enters an ADJUST_PSTT state. In this state, the
6 PS_TT[8:0] signal is set equal to the result R[8:0] provided
7 by bias adder/subtractor 1532. The "A" input terminal of
8 comparator 1533 is coupled to receive the result R[8:0] of
9 bias adder/subtractor 1532. The "B" input terminal of
10 comparator 1533 is coupled to receive the PS_TT[8:0] signal
11 from up/down counter 1311. Comparator 1533 provides logic
12 high signal to the clock enable input terminal (CE) of
13 up/down counter 1311 if the R[8:0] signal is not equal to the
14 PS_TT[8:0] signal, thereby enabling up/down counter 1311.
15 Comparator 1533 also provides a logic high signal to the UP
16 terminal of up/down counter 1311 if the PS_TT[8:0] signal is
17 less than the R[8:0] signal. Conversely, logic block 1533
18 provides a logic low signal to the UP terminal of up/down
19 counter 1311 if the PS_TT[8:0] signal is greater than or
20 equal to the R[8:0] signal. Up/down counter 1311 is
21 incremented by one if the UP terminal receives a logic "1"
22 signal, and is decremented by one if the UP terminal receives
23 a logic "0" signal. Upon RESET, up/down counter 1311 is
24 reset to a logic zero value. Consequently, up/down counter
25 1311 will adjust the PS_TT[8:0] value until this value is
26 equal to the R[8:0] value (or until overflow occurs). When
27 the PS_TT[8:0] signal is equal to the R[8:0] signal, or
28 OVERFLOW is true, control logic 1550 sets the DONE signal.
29 Control logic 1550 sets the PS_LOCKED signal the first time
30 the DONE signal is asserted after RESET. The PS_LOCKED
31 signal informs DLL control 1312 that the phase shift logic
32 1310 has been properly initialized. An exception is during
33 the locking process, where the DONE signal is not set if
34 OVERFLOW is true, so the PS_LOCKED signal will not set
35 either.

1 The primary state machine then enters the WT_GONOT
2 state, in which the state machine waits until the GO signal
3 is inactive, and then enters the IDLE state.

4 External user interface 1320 is provided to allow
5 dynamic changes to the phase shift (i.e., the fraction
6 numerator of equations (8), (9), (10)) during normal operation.
7 The user may increment or decrement the fraction numerator.
8 In the variable modes, the fraction numerator may cross from
9 positive through zero to negative, and vice-versa. External
10 user interface 1320 is synchronous with the external clock
11 signal PSCLK. Note that the rest of the clocked elements of
12 phase shift control circuit 1310 are clocked by the
13 PS_DLY_OUT clock signal. The module uses standard techniques
14 to cross from the PSCLK domain to the PS_DLY_OUT clock domain
15 and vice-versa.

16 A second state machine, namely, a dynamic phase change
17 (DPS) state machine, controls user interface 1320. Running
18 in the PS_DLY_OUT clock domain, the DPS state machine also
19 ensures that changes are made to the signed up/down counter
20 1511 only when this counter is not being used by the primary
21 state machine. After the DPS state machine is started,
22 control proceeds to the next state after one clock cycle,
23 unless otherwise noted in the description below.

24 The DPS state machine starts in a DPS_IDLE state,
25 wherein the state machine waits for the PSEN signal to be
26 asserted. Upon detecting the PSEN signal (as clocked in by
27 the PSCLK signal), the DPS state machine will proceed to the
28 next state. However, to avoid logical discontinuity, the
29 DPS_IDLE state cannot be exited unless the primary state
30 machine is in the IDLE state.

31 The DPS state machine then enters a DPS_CNTR state, in
32 which counter 1511 is incremented if the captured PSINCDEC
33 signal has a logic "1" value, and decremented if the captured
34 PSINCDEC signal has a logic "0" value.

35 The DPS state machine then enters a DPS_W_ADJUST_PSTT
36 state, in which the DPS state machine waits until the primary

1 state machine completes a corresponding adjustment cycle.
2 The DPS state machine remains in this state until the exit
3 conditions for the primary state ADJUST_PSTT are true.

4 The DPS state machine then enters a DPS_SETDONE state,
5 in which the PSDONE signal (which is coupled to the PSCLK
6 domain) is set. The asserted PSDONE signal informs the user
7 that the requested phase change has been completed.
8 Processing then returns to the DPS_IDLE state.

9 In the foregoing manner, digital phase shifter 1100 is
10 capable of precisely modifying the delay between the REF_CLK
11 and S_CLK signals. This operation is controlled to
12 compensate for low frequency and high frequency signals, as
13 well as for variations in temperature. Advantageously, the
14 phase can be modified both automatically and under user
15 control.

16 Fig. 16 is a block diagram of a digital spread spectrum
17 (DSS) circuit 1600 in accordance with another embodiment of
18 the present invention. DSS circuit 1600 is coupled between
19 up/down counter 1311 and binary-to-gray conversion circuit
20 1305 (See, Fig 13). Both up/down counter 1311 and DSS
21 circuit 1600 are clocked by the PS_DLY_OUT signal. In
22 general, DSS circuit 1600 can be configured to add or
23 subtract small predetermined values to the PS_TT[8:0] signal
24 to create the JOINT_TT[8:0] signal. In this case, DSS
25 circuit 1600 is in a spread spectrum mode. While in the
26 spread spectrum mode, the JOINT_TT[8:0] signal causes small
27 variations in the clock period of the PS_DLY_OUT signal.

28 DSS circuit 1600 can also be configured to disable the
29 spread spectrum mode, such that the PS_TT[8:0] signal is
30 transmitted through DSS circuit 1600 without change. This is
31 referred to as the DSS bypass mode.

32 Fig. 17A illustrates the DSS bypass mode, in which the
33 PS_DLY_OUT signal has a frequency of 100 MHz and an energy of
34 X. In determining the electromagnetic emission of the chip,
35 the FCC will use a window 1701 having a specified width
36 (e.g., 1 MHz). Because the PS_DLY_OUT signal has a single

1 frequency, the 1 MHz window 1701 will capture all of the
2 energy X of the PS_DLY_OUT signal.

3 Fig. 17B illustrates one of the spread spectrum modes of
4 DSS circuit 1600, in which the PS_DLY_OUT signal is
5 controlled to exhibit frequencies of approximately 98, 98.5,
6 99, 99.5, 100, 100.5, 101, 101.5 and 102 MHz. Because each
7 of these frequencies exists only 1/9 of the time, the
8 PS_DLY_OUT signal has an energy of 1/9 X at each of these
9 frequencies. As a result, the 1 MHz window 1701 will only
10 capture three of the nine frequencies at any given time. As
11 a result, the energy detected by the 1 MHz window has a
12 maximum value of 1/3 X. This represents a significant
13 reduction in EMI energy versus the bypass mode. As the base
14 frequency of the PS_DLY_OUT signal changes, the number of
15 frequencies existing within the 1 MHz window may change.

16 Fig. 18 is a circuit diagram of DSS circuit 1600 in
17 accordance with one embodiment of the present invention. DSS
18 circuit 1600 includes pattern generator 1801, signed adder
19 1802, pipeline register 1803, 5-bit up counter 1804, D flip-
20 flops 1805-1806, inverter 1807, OR gate 1808, and NAND gate
21 1809. The elements of DSS circuit 1600 are clocked by the
22 PS_DLY_OUT signal.

23 To enable DSS circuit 1600, a configuration memory bit
24 is programmed to store a logic low EN_DSS# signal, thereby
25 enabling the user to select whether or not the DSS circuit
26 1600 can be used during normal operation of delay lock loop
27 400. If the user wants to use DSS circuit 1600, the user
28 must also provide a logic high DSS_EN signal to NAND gate
29 1809. The RESET signal is then asserted high, which causes
30 OR gate 1808 to provide a logic high signal to clear pattern
31 generator 1801, clear flip-flop 1805, and asynchronously set
32 5-bit up counter 1804 to a value of "10000" (i.e., binary
33 16).

34 Initially, delay lock loop 400 is not locked (i.e.,
35 DLL_LOCKED is low). As a result, NAND gate provides a logic
36 "1" value, which is loaded into flip-flop 1805. This logic

1 "1" value causes OR gate 1808 to provide a logic "1" output
2 value, thereby maintaining pattern generator 1801 in a
3 cleared state. At this time, pattern generator 1801 provides
4 a trim signal t[9:0] having a value of zero. As a result,
5 the PS_TT[8:0] signal provided by up/down counter 1311 passes
6 through adder 1802 and pipeline register 1803 unchanged.
7 This effectively removes DSS circuit 1600 from delay lock
8 loop 400.

THIS IS BY
9 After delay lock loop 400 becomes locked, the
10 DLLS_LOCKED signal transitions to a logic high value. In
11 response, NAND gate 1809 provides a logic "0" output signal.
12 This logic "0" signal is latched into flip-flop 1805 on the
13 next rising edge of the PS_DLY_OUT signal. As a result, OR
14 gate 1808 provides a logic "0" signal which releases pattern
15 generator 1801, 5-bit up counter 1804 and flip-flop 1805. At
16 this time, pattern generator 1801 is enabled to generate a
17 predetermined pattern. The particular pattern is selected by
18 the SPREADSEL[3:0] signal. In one embodiment, pattern
19 generator 1801 is capable of generating patterns for creating
20 spread-2, spread-4, spread-6, spread-8, spread-16, spread-32,
21 and spread-64 configurations. In the present example, the
22 SPREADSEL[3:0] signal is selected to provide a pattern for a
23 spread-8 configuration.

24 In the described embodiment, up/down counter 1311
25 asserts a CHANGE_PS signal when the contents of up/down
26 counter 1311 are being changed. When the CHANGE_PS signal is
27 asserted, the output t[9:0] of pattern generator 1801 is
28 prevented from changing, thereby avoiding contention between
29 pattern generator 1801 and up/down counter 1311.

30 In general, pattern generator provides a trim value
31 t[9:0] to the A input terminal of adder 1802, and counter
32 1311 provides the PS_TT[8:0] signal to the B input terminal
33 of adder 1802. Note that a PS_TT[9] bit having a logic "0"
34 value is concatenated to the PS_TT[8:0] signal. Also note
35 that bits t[9:4] are set equal to t[3]. In the spread-8

1 pattern, the trim value $t[9:0]$ can have values of 0, +1, +2,
2 -1 and -2.

3 Adder 1802 performs a signed addition of the $PS_TT[9:0]$
4 and $t[9:0]$ signals, thereby providing a sum signal $s[9:0]$.
5 As long as the sum bit $s[9]$ has a logic "0" value, the sum
6 value $s[8:0]$ will be within the operating range of 512-tap
7 delay line 1304. The logic "0" sum bit $s[9]$ enables pipeline
8 register 1803 to latch the $s[8:0]$ signal. Pipeline register
9 1803 then transmits the $s[8:0]$ to binary-to-gray decoder 1305
10 as the $PS_TAP_TRIM[8:0]$ signal.

11 If sum bit $s[9]$ has a value of "1", then pipeline
12 register 1803 is disabled, and does not latch the
13 corresponding $s[8:0]$ signal. The logic "1" $s[9]$ bit is
14 clocked into flip-flop 1805 by the PS_DLY_OUT signal and
15 provided to the synchronous input terminal $SI(6)$ of 5-bit up
16 counter 1804. When the PS_DLY_OUT signal is asserted high,
17 5-bit up counter 1804 is set to a value of "00110".

18 It should be noted that prior to loading counter 1804
19 with a value of "00110", counter 1804 was set to a value of
20 "10000". Bit [4] of the counter (i.e., the "1" bit in
21 "10000") is provided to inverter 1807. Inverter 1807, in
22 turn, provides a logic "0" bit the enable input (CE) of 5-bit
23 up counter 1804. As a result, the 5-bit up counter 1804 is
24 effectively disabled until it is loaded with the "00110"
25 value. The output of inverter 1807 is also used as a
26 $DSS_OVERFLOW$ signal.

27 After the "00110" value has been loaded into 5-bit up
28 counter 1804, bit [4] of the counter has a logic "0" state.
29 As a result, the $DSS_OVERFLOW$ signal is asserted high, and 5-
30 bit up counter 1804 is enabled. Once the $DSS_OVERFLOW$ signal
31 is set at a logic high state, this signal will remain in a
32 logic high state for at least 10 cycles of the PS_DLY_OUT
33 signal. The $DSS_OVERFLOW$ signal will be reset to a logic low
34 value if the $s[9]$ signal transitions back to a logic "0"
35 value, and remains at a logic "0" value for 10 consecutive
36 cycles of the PS_DLY_OUT signal. After the $s[9]$ signal

1 transitions to a logic "0" value, counter 1804 will begin
2 counting up. If the s[9] signal remains at a logic "0" state
3 for ten clock cycles, then counter 1804 will count up to
4 "10000", thereby clearing the DSS_OVERFLOW signal. However,
5 if the s[9] signal has a logic "1" value any time during
6 these ten clock cycles, then the 5-bit counter will be reset
7 to a "00110" value, thereby resulting in a logic high
8 DSS_OVERFLOW signal. This enables the user to reliably
9 sample the status of the DSS_OVERFLOW signal. (In the
10 described embodiment, the DSS_OVERFLOW signal is logically
11 OR'ed with the OVERFLOW signal of Fig. 15, thereby providing
12 a single signal to identify overflow conditions.)

13 The trim signals t[9:0] generated by pattern generator
14 1801 determine how the period of the PS_DLY_OUT signal is
15 adjusted. A spread-8 configuration will now be described.
16 This example assumes that the REF_CLK signal has a base
17 frequency of 100 MHz, and that the 512-tap/trim delay line
18 1304 is connected in line with the REF_CLK signal (i.e.,
19 S_LAGS_REF = 1 and CENTERED = 1). This example also assumes
20 that one trim delay is equal to 50 picoseconds (ps).

21 Figs. 19A-19B are waveform diagrams illustrating the
22 REF_CLK and S_CLK signals when DSS circuit 1600 is controlled
23 to implement a spread-8 configuration. During the first
24 three clock cycles C1-C3 (i.e., until 30000 ns) DSS circuit
25 1600 is disabled by the DSS_EN signal provided by the user.
26 As a result, pattern generator 1801 provides a trim signal
27 t[9:0] having a value of 0. During these three clock cycles
28 C1-C3, the REF_CLK and S_CLK are synchronized, with each of
29 these signals having a period of 10000 ns.

30 During the fourth clock cycle C4, pattern generator 1801
31 is enabled and provides a trim signal t[9:0] having a value
32 equal to 2 trim settings. In response, signed adder circuit
33 1802 provides adds two trim settings to the PS_TT[8:0] signal
34 provided by up/down counter 1311. As a result, the cycle C4
35 of the S_CLK signal is lengthened by two trim settings, or
36 100 ps. Cycle C4 of the S_CLK signal therefore has a period

1 of 10100 ns, such that the rising edge of the fifth cycle C5
2 of the S_CLK signal occurs at 40100 ns. At this time, there
3 is an initial offset of 100 ns between the REF_CLK and S_CLK
4 signals. This initial offset is provided one time only
5 before implementing the normal spread-8 configuration. This
6 initial offset enables the spread 8 configuration to be
7 implemented in an optimal manner as described below.

8 During the fifth clock cycle C5, pattern generator 1801
9 provides a trim signal $t[9:0]$ having a value equal to 1 trim
10 setting. This represents a change of -1 trim setting (-50
11 ps) with respect to the previous cycle. As a result, the
12 S_CLK signal will have a period equal to 10000 ps - 50 ps, or
13 9950 ps, such that the rising edge of the next clock cycle C6
14 occurs at 50050 ns. The frequency of the S_CLK signal is
15 equal to 100.5 MHz during cycle C5.

16 During the sixth clock cycle C6, pattern generator 1801
17 provides a trim signal $t[9:0]$ having a value equal to 2 trim
18 settings. This represents a change of +1 trim setting (+50
19 ps) with respect to the previous cycle. As a result, the
20 S_CLK signal will have a period equal to 10000 ps + 50 ps, or
21 10050 ps, such that the rising edge of the next clock cycle
22 C7 occurs at 60100 ns. The frequency of the S_CLK signal is
23 equal to 99.5 MHz during cycle C5.

24 Processing continues, with pattern generator 1801
25 providing trim signals $t[9:0]$ having values of 0, 2, -1, 2, -
26 2, 2, and 2 during clock cycles C7-C13, respectively. These
27 trim values correspond with trim differences of -2, 2, -3, 3,
28 -4, 4 and 0 during clock cycles C7-C13, respectively. As a
29 result, the S_CLK signal has periods of 9900, 10100, 9850,
30 10150, 9800, 10200 and 10000 ps during clock cycles C7-C13,
31 respectively. This means that the S_CLK signal has
32 frequencies of 101, 99, 101.5, 98.5, 102, 98 and 100 MHz,
33 during clock cycles C7-C13, respectively. This provides for
34 each of the eight frequencies illustrated in Fig. 17B, plus
35 the base frequency of 100 MHz. The pattern of clock cycles

09564523-100500

1 C5-C13 is repeated during the operation of delay lock loop
2 400, thereby continuing the spread spectrum operation.

3 In the example of Figs. 19A and 19B, the skew between
4 the REF_CLK signal and the S_CLK signal is equal to 50, 100,
5 0, 100, -50, 100, -100, 100 and 100 ps during clock cycles
6 C5-C13, respectively. Thus, the skew between these two
7 signals has a maximum value of 100 ps, and an average value
8 of 44.4 ps. This is a relatively stable clock signal in view
9 of the large number of frequencies provided.

10 In another embodiment of the present invention, the
11 initial offset provided during clock cycle C4 can be
12 eliminated. In this embodiment, the trim values provided by
13 pattern generator 1801 are selected to be -1, 0, -2, 0, -3,
14 0, -4, 0 and 0 during clock cycles C5-C13, respectively.
15 These trim values correspond with trim differences of -1, 1,
16 -2, 2, -3, 3, -4, 4 and 0 during clock cycles C5-C13,
17 respectively. As a result, the S_CLK signal has periods of
18 9950, 10050, 9900, 10100, 9850, 10150, 9800, 10200 and 10000
19 ps during clock cycles C5-C13, respectively. This means that
20 the S_CLK signal has frequencies of 100.5, 99.5, 101, 99,
21 101.5, 98.5, 102, 98 and 100 MHz, during clock cycles C5-C13,
22 respectively. Again, this provides for each of the eight
23 frequencies illustrated in Fig. 17B, plus the base frequency
24 of 100 MHz. The pattern of clock cycles C5-C13 is repeated
25 during the operation of delay lock loop 400, thereby
26 continuing the spread spectrum operation. In this example,
27 the skew between the REF_CLK signal and the S_CLK signal is
28 equal to -50, 0, -100, 0, -150, 0, -200, 0, and 0 ps,
29 respectively during clock cycles C5-C13, respectively. Thus,
30 the skew between these two signals has a maximum value of 200
31 ps, and an average value of -55.6 ps.

32 Although a spread-8 configuration (i.e., a base
33 frequency plus eight spread frequencies) has been described,
34 it is understood that other spread spectrum configurations
35 can also be implemented, and are considered to fall within

the scope of the present invention. For example, spread-2, spread-4 and spread-6 configurations can be provided. Pattern generator 1801 provides a pattern of trim values equal to -1, 0 and 0 to provide the three frequencies of the spread-2 configuration.

Pattern generator 1801 provides a pattern of trim values equal to -1, 0, -2, 0 and 0 to provide the five frequencies of the spread-4 configuration. To add an initial offset, pattern generator 1801 can provide a pattern of trim values equal to 1 (initial offset), 0, 1, -1, 1, and 1 to provide the five frequencies of the spread-4 configuration.

Pattern generator 1801 provides a pattern of trim values equal to -1, 0, -2, 0, -3, 0 and 0 to provide the seven frequencies of the spread-6 configuration. To add an initial offset, pattern generator 1801 can provide a pattern of trim values equal to 1 (initial offset), 0, 1, -1, 1, -2, 1 and 1 to provide the seven frequencies of the spread-6 configuration.

Table 1 below summarizes characteristics of spread-2, spread-4, spread-6 and spread-8 configurations for a 100 MHz base clock signal.

TABLE 1

DSS MODE	NONE	SPREAD-2	SPREAD-4	SPREAD-6	SPREAD-8
# of new freq.	0	2	4	6	8
Total # of freq.	1	3	5	7	9
% EMI peak reduction	0	67	80	86	89
+/- spread (trim units)	0	1	2	3	4
+/- spread (ps)	0	50	100	150	200

Table 2 below summarizes the +/- spread of the base clock signal for the various DSS modes for selected frequencies between 25 MHz and 400 MHz.

TABLE 2

Freq.	Period	+/- Spread (% of period)				
		NONE	SPRD-2	SPRD-4	SPRD-6	SPRD-8
25 MHz	40 ns	0.00%	0.13%	0.25%	0.38%	0.50%
50	20	0.00%	0.25%	0.50%	0.75%	1.00%
100	10	0.00%	0.50%	1.00%	1.50%	2.00%
200	5	0.00%	1.00%	2.00%	3.00%	4.00%
400	2.5	0.00%	2.00%	4.00%	6.00%	8.00%

Table 3 below summarizes the number of ideal peaks within a 1 MHz window for the various DSS modes for selected frequencies between 25 MHz and 400 MHz.

TABLE 3

Freq.	Period	# IDEAL PEAKS INSIDE 1 MHZ WINDOW				
		NONE	SPRD-2	SPRD-4	SPRD-6	SPRD-8
25 MHz	40 ns	1	3	5	7	9
50	20	1	3	5	7	9
100	10	1	3	3	3	3
200	5	1	1	1	1	1
400	2.5	1	1	1	1	1

Table 4 below summarizes the percentage of EMI energy reduction within a 1 MHz window for the various DSS modes for selected frequencies between 25 MHz and 400 MHz.

TABLE 4

Freq.	Period	+/- Spread (% of period)				
		NONE	SPRD-2	SPRD-4	SPRD-6	SPRD-8
25 MHz	40 ns	0%	0%	0%	0%	0%
50	20	0%	0%	0%	0%	0%
100	10	0%	0%	40%	57%	67%
200	5	0%	67%	80%	86%	89%
400	2.5	0%	67%	80%	86%	89%

Table 5 below summarizes the EMI energy reduction within a 1 MHz window in db for the various DSS modes for selected frequencies between 25 MHz and 400 MHz.

TABLE 5

Freq.	Period	1 MHZ EMI ENERGY REDUCTION (db)				
		NONE	SPRD-2	SPRD-4	SPRD-6	SPRD-8
25 MHz	40 ns	0	0	0	0	0
50	20	0	0	0	0	0
100	10	0	0	-2.2	-3.7	-4.8
200	5	0	-4.8	-7.0	-8.5	-9.5
400	2.5	0	-4.8	-7.0	-8.5	-9.5

Table 6 below summarizes the 1 MHz window range for selected frequencies between 25 MHz and 400 MHz.

TABLE 6

Frequency (MHz)	Period (ns)	1 MHz Window Range (period)
25	40	39.22 to 40.82 ns
50	20	19.80 to 20.20 ns
100	10	9.95 to 10.05 ns
200	5	4.99 to 5.01 ns
400	2.5	2.497 to 2.503 ns

Although the invention has been described in connection with several embodiments, it is understood that this invention is not limited to the embodiments disclosed, but is capable of various modifications which would be apparent to a person skilled in the art. For example, in view of this disclosure those skilled in the art can define other clock phase shifters, delay lines, output generators, controllers, phase detectors, and so forth, and use these alternative features to create a method, circuit, or system according to the principles of this invention. Thus, the invention is limited only by the following claims.

APPENDIX A

APPENDIX A CORRESPONDING TERMINOLOGY
 TERMINOLOGY USED IN THE SPECIFICATION

DLL	Delay Lock Loop 400
ZD2 section	Clock Phase Shifter 350
PS section	Digital Phase Shifter 1100
ZD1 section	Delay Line 310

ResetZD2	Reset Clock Phase Shifter 350
ResetPS	Reset Digital Phase Shifter 1100
ResetZD1	Reset Delay Line 310
ChangeZD2	When asserted, allows Clock Phase Shifter 350 to make a tap/trim change
GoPS	GO (when asserted, allows Digital Phase Shifter to make a tap/trim change)
ChangeZD1	When asserted, allows Delay Line 310 to make a tap/trim change
DLL_locked	DLL_LOCKED
locked_pre	Similar to DLL_LOCKED, except it remains true even if overflow occurs
PSlocked	When asserted, indicates that the initial tap/trim for delay line 1304 has been set
FREEZEDLL	User signal that holds all updates to delay lines in Delay Lock Loop 400
EnZD2	Configuration bit that enables Clock Phase Shifter 350 (allows ResetZD2 to go inactive)
EnPS	Configuration bit that enables Digital Phase Shifter 1100 (allows ResetPS to go inactive)
EnZD1	Configuration bit that enables Delay Line 310 (allows Reset ResetZD1 to go inactive)

00684523-100500

CTLMODE	When true, DLL 400 is in test mode
Z2locked	When asserted, indicates that Clock Phase Shifter 350 delay lines have been adjusted (i.e., total delay is one CLKIN period)
DonePS	DONE
Z1locked	When asserted, indicates that Delay line 310 has been adjusted (i.e., PS_S_CLK and PS_REF_CLK are in phase)
ZD2overflow	When asserted, indicates a wrap attempt on one or more delay line sections in Clock Phase Shifter 350
ZD1overflow	When asserted, indicates a wrap attempt on Delay Line 310
Clk	REF_CLK

0963433-100600

```

/*****
*
*   File Name:   DLLmngr_9.v
*   Version:    snapshot_00.08.07revA
*   Generated:   Tue Sep 26 17:15:26 2000 (Pacific Time)
*
*   Author:     John Logue
*   Email:      john.logue@xilinx.com
*   Company:    Xilinx
*
*               Copyright (c) 2000 Xilinx, Inc.
*               All rights reserved
*
*****/

`timescale 1 ps / 1 ps

module DLLmngr_9(
    ResetZD2, ResetPS, ResetZD1,
    ChangeZD2, GoPS, ChangeZD1,
    DLL_locked, locked_pre,
    PSlocked, FREEZEDLL, EnZD2, EnPS, EnZD1, CTLMODE,
    DeadTime, LiveTime, z2Locked, DonePS, z1Locked,
    ZD2overflow,
    // PSoverFlow,
    ZD1overflow,
    Clk, Reset);

// This is a one-hot state machine that controls modules zd2Synth, PhaseShift,
// and zd1Synth.
// It has the following configuration:
//   Timeout counter width in bits: 9

// FREEZEDLL operation:

// If FREEZEDLL_S goes true when in a state between updates (deadtime), that
// state is retained until FREEZEDLL_S goes false; i.e., it hangs in that state
// until FREEZEDLL_S goes false.

// If FREEZEDLL_S goes true when updating PS, nothing happens until it advances
// to the next deadtime state, where it hangs until FREEZEDLL_S goes false.

// If FREEZEDLL_S goes true when updating ZD1 or ZD2, it immediately exits to
// the corresponding deadtime state, where it hangs until FREEZEDLL_S goes
// false.

output ResetZD2;    // ZD2 reset
output ResetPS;     // PhaseShift reset
output ResetZD1;    // ZD1 reset
output ChangeZD2;    // Allow ZD2 to make a tap/trim change
output GoPS;         // Go signal to PhaseShift
output ChangeZD1;    // Allow ZD1 to make a tap/trim change
output DLL_locked;   // DLL top level locked status
output locked_pre;   // locked status that remains true if overflow occurs

```



```

reg [WLZD1:0] NextState; // next state

assign ChangeZD2 = State[IDLE] | State[WLZD2];
assign ChangeZD1 = State[INITZD1] | State[WLZD1];
assign GoPS      = State[INITPS] | State[WDONE];

// ***** Resets *****

// This section generates reset signals for ZD2, PS, ZD1, and the DLLmngr
// itself. All resets go active while Reset is true. The various resets
// then go inactive (or not) as follows:

// ResetZD2 - goes inactive immediately if EnZD2 is true; otherwise,
//            it never goes inactive.

// ResetPS - goes inactive immediately if EnPS & CTLMODE are both true, else
//            goes inactive when ZD2 locks, if EnPS is true; otherwise,
//            it never goes inactive.

// ResetZD1 - goes inactive immediately if EnZD1 & CTLMODE are both true;
//            otherwise, it goes inactive synchronous to the rising edge of
//            Clk when signal RelResetZD1 is true.

// DLLmngrReset - goes inactive immediately if any of ZD2, PS, or ZD1 are
//                enabled AND Test Mode is not selected. Otherwise, it
//                never goes inactive.

wire ResetZD2n = ~(Reset | (~(ResetZD2n | EnZD2))); // SR latch
wire ResetZD2  = ~ResetZD2n;

wire ClrResetPS = EnPS & (z2Locked | CTLMODE); // JDL 07/31/00

wire ResetPSn = ~(Reset | (~(ResetPSn | ClrResetPS))); // SR latch JDL
07/24/00
wire ResetPS  = ~ResetPSn;

wire RelResetZD1 = State[INITPS] & EnZD1 & (~EnPS | DonePS_S);

wire AsyncClr = EnZD1 & CTLMODE;
reg ResetZD1;
always @(posedge Clk or posedge Reset or posedge AsyncClr)
begin
    if (Reset)      ResetZD1 <= #`FFDLY 1'b1; // async set
    else if (AsyncClr) ResetZD1 <= #`FFDLY 1'b0; // async clr
    else if (RelResetZD1) ResetZD1 <= #`FFDLY 1'b0; // edge trigger
end

wire EnDLLmngr = (EnZD2 | EnPS | EnZD1) & ~CTLMODE;

wire DLLmngrResetN = ~(Reset | (~(DLLmngrResetN | EnDLLmngr))); // SR latch
wire DLLmngrReset  = ~DLLmngrResetN;

// ***** TimeOut Counter *****

wire Sellive = State[WDZD2] | State[WDPS] | State[WDZD1];

wire [7:0] TOcntrMux = Sellive ? LiveTime : DeadTime;

```

```

wire [8:0] TOcntr;
wire TO = TOcntr[8]; // timeout

wire LdTOcntr = State[INITZD1]
               | (State[WDZD2] & TO)
               | (State[WLZD2] & TO)
               | (State[WDPS] & TO)
               | State[WDONE]
               | (State[WDZD1] & TO)
               | (State[WLZD1] & TO);

// instantiate timeout counter
cntr_9_AI_0_LD_CE_DN TOC(
    .Out(TOcntr),
    .AI(DLLmngrReset),
    .LD(LdTOcntr),
    .In({1'b0, TOcntrMux}),
    .CE(~TO),
    .Clk(Clk));

// ***** State Machine *****

always @(posedge Clk or posedge DLLmngrReset)
    if (DLLmngrReset) State <= #`FFDLY 1; // select IDLE state on Reset
    else State <= #`FFDLY NextState;

// next state network
always @(State or EnZD1 or EnZD2 or EnPS or
    z2Locked_S or DonePS_S or z1Locked_S or TO or FREEZEDLL_S)
begin
    NextState = 0;

    case(1'b1) // synopsys parallel_case full_case
        State[IDLE] : begin // idle
            if (EnZD2 & ~z2Locked_S) NextState[IDLE] = 1'b1;
            else NextState[INITPS] = 1'b1;
        end
        State[INITPS] : begin // initialize PhaseShift
            if (EnPS & ~DonePS_S) NextState[INITPS] = 1'b1;
            else NextState[INITZD1] = 1'b1;
        end
        State[INITZD1] : begin // initialize ZD1
            if (EnZD1 & ~z1Locked_S)
                begin
                    NextState[INITZD1] = 1'b1;
                end
            else
                begin
                    if (EnZD2) NextState[WDZD2] = 1'b1;
                    else NextState[WDZD1] = 1'b1;
                end
            end
        end
        State[WDZD2] : begin // wait dead time before changing ZD2
            if (~TO | FREEZEDLL_S) NextState[WDZD2] = 1'b1;
            else NextState[WLZD2] = 1'b1;
        end
    end
end

```

```

State[WLZD2] : begin // wait live time while changing ZD2
               if (~TO & ~FREEZEDLL_S) NextState[WLZD2] = 1'b1;
               else if (FREEZEDLL_S) NextState[WDZD2] = 1'b1;
               else if (EnPS) NextState[WDPS] = 1'b1;
               else if (EnZD1) NextState[WDZD1] = 1'b1;
               else NextState[WDZD2] = 1'b1;
               end
State[WDPS] : begin // wait dead time before changing PhaseShift
               if (~TO | FREEZEDLL_S) NextState[WDPS] = 1'b1;
               else NextState[WDONE] = 1'b1;
               end
State[WDONE] : begin // wait PhaseShift Done
               if (~DonePS_S) NextState[WDONE] = 1'b1;
               else if (EnZD1) NextState[WDZD1] = 1'b1;
               else NextState[WDZD2] = 1'b1;
               end
State[WDZD1] : begin // wait dead time before changing ZD1
               if (~TO | FREEZEDLL_S) NextState[WDZD1] = 1'b1;
               else NextState[WLZD1] = 1'b1;
               end
State[WLZD1] : begin // wait live time while changing ZD1
               if (~TO & ~FREEZEDLL_S) NextState[WLZD1] = 1'b1;
               else if (FREEZEDLL_S) NextState[WDZD1] = 1'b1;
               else if (EnZD2) NextState[WDZD2] = 1'b1;
               else NextState[WDZD1] = 1'b1;
               end
default : begin
               NextState = 9'bx;
               end
endcase
end
endmodule

```

DLL Manager

Overview

This is the master control module for the DLL. It performs the following functions:

1. Sequences the Reset signals to the ZD2, PS, and ZD1 sections.
2. Ensures that the sections initialize (lock) in the proper sequence (ZD2, PS, ZD1).
3. Ensures that clock cycles that are shortened or lengthened due to a trim change are not sampled by the phase detector of another DLL section. This is done by allowing only one section to change trims at a given time, and inserting a no activity time between each allowed change.
4. Generates DLL_locked signal.

FREEZEDLL Notes

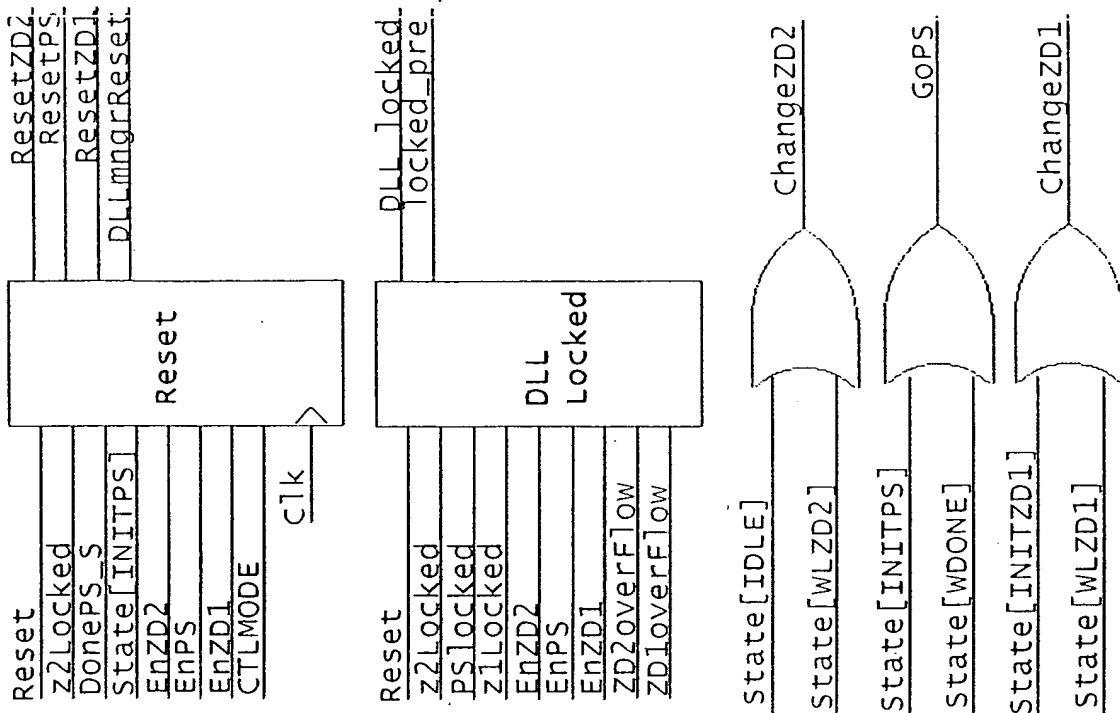
If FREEZEDLL_S goes true when in a state between updates (deadtime), that state is retained until FREEZEDLL_S goes false; i.e., it hangs in that state until FREEZEDLL_S goes false.

If FREEZEDLL_S goes true when updating PS, nothing happens until it advances to the next deadtime state, where it hangs until FREEZEDLL_S goes false.

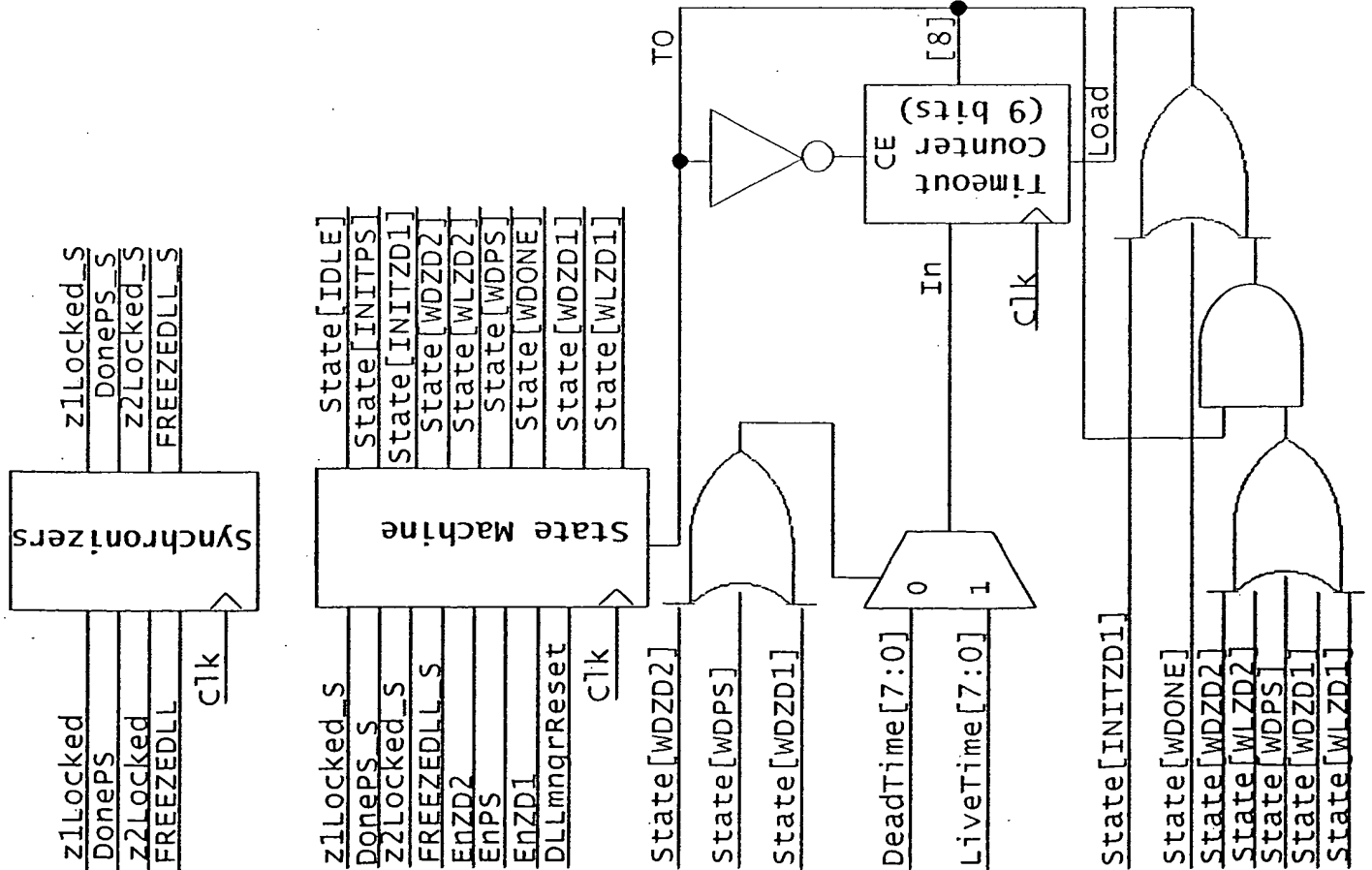
If FREEZEDLL_S goes true when updating ZD1 or ZD2, it immediately exits to the corresponding deadtime state, where it hangs until FREEZEDLL_S goes // false.

Input & Output Pins

Outputs	
ResetZD2	ZD2 reset
ResetPS	Phaseshift reset
ResetZD1	ZD1 reset
ChangeZD2	Allow ZD2 to make a tap/trim change
GOPS	Go signal to Phaseshift
ChangeZD1	Allow ZD1 to make a tap/trim change
DLL_locked	DLL top level locked status
Locked_pre	Locked status that remains true if overflow occurs
Inputs	
PSlocked	The initial Phase Shift tap/trim has been set
FREEZEDLL	Halts all DLL delay line updates as soon as possible
EnZD2	config bit: ZD2 enabled
EnPS	config bit: Phaseshift enabled
EnZD1	config bit: ZD1 enabled
CTLMODE	true when test mode selected
DeadTime [7:0]	init value for Timeout cntr between module enables
LiveTime [7:0]	init value for Timeout cntr during ZD2 & ZD1 enables
z2Locked	ZD2 is locked
DonePS	Done signal from Phaseshift
z1Locked	ZD1 locked
ZD2overflow	wrap attempt on one or more ZD2 delay lines
ZD1overflow	wrap attempt on ZD1 delay line
Reset	Asynchronous DCM Reset (RST)
Clk	Connected to DCM input clock (CLKIN)



Block Diagram

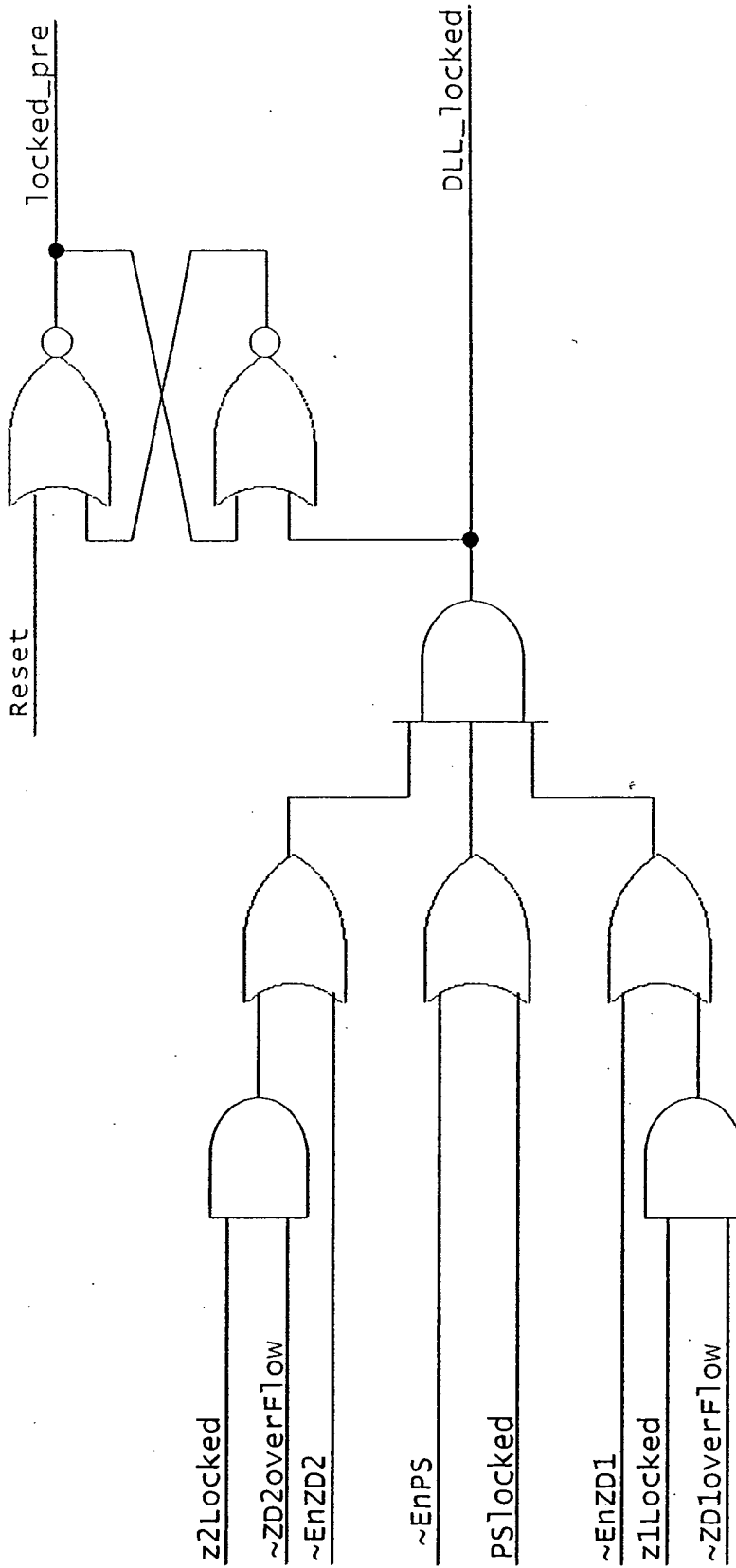


000001" B2643950

Reset Notes

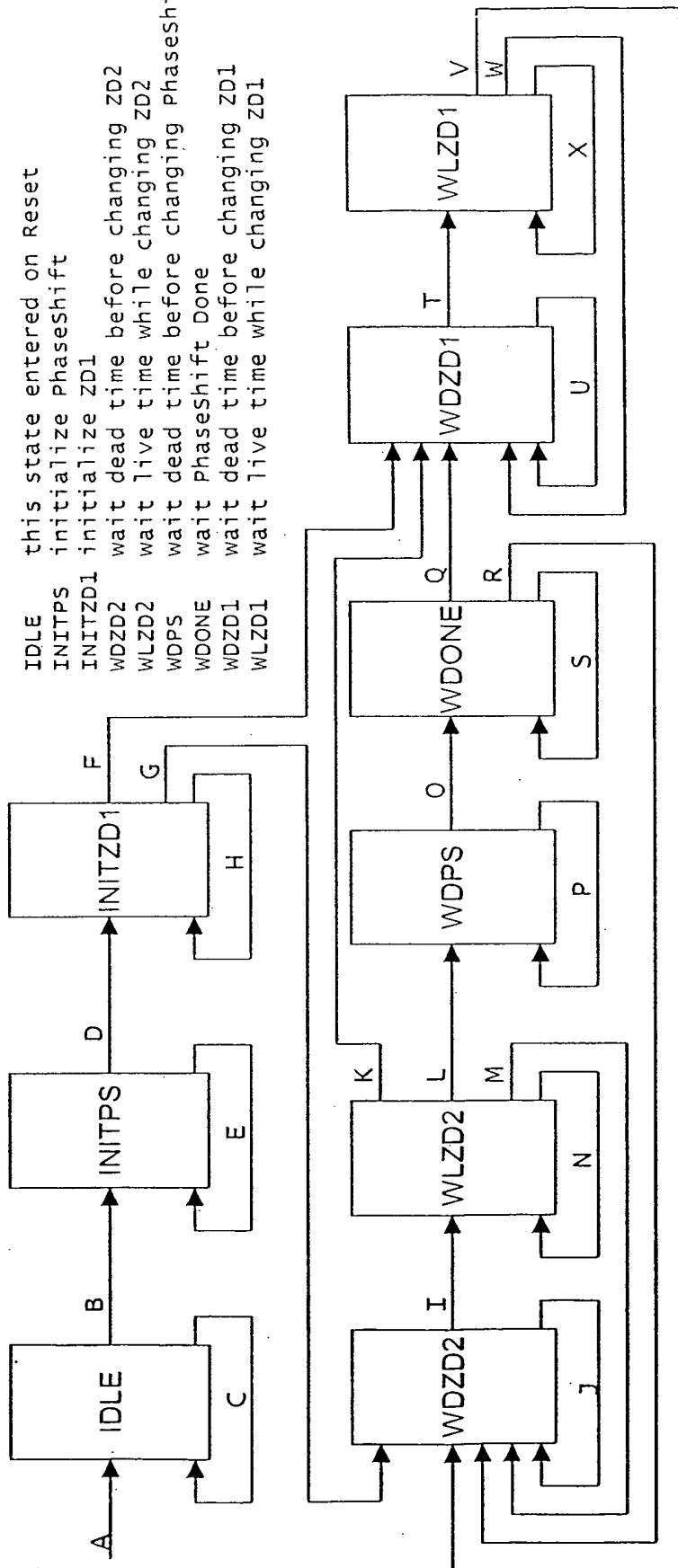
This section generates reset signals for ZD2, PS, ZD1, and the DLLmngR itself. All resets go active while Reset is true. The various resets then go inactive (or not) as follows:

- ResetZD2 - goes inactive immediately if EnZD2 is true; otherwise, it never goes inactive.
- ResetPS - goes inactive immediately if EnPS & CTLMODE are both true, else goes inactive when ZD2 locks, if EnPS is true; otherwise, it never goes inactive.
- ResetZD1 - goes inactive immediately if EnZD1 & CTLMODE are both true; otherwise, it goes inactive synchronous to the rising edge of CLK when signal ReResetZD1 is true.
- DLLmngRReset - goes inactive immediately if any of ZD2, PS, or ZD1 are enabled AND Test Mode is not selected. Otherwise, it never goes inactive.



DLL_locked goes true when all selected functions are locked. Overflow on ZD1 or ZD2 causes it to go false (if the function is selected). It is always true if no functions are selected.

DLL Locked



Conditions

A DLLmngReset
 B ~C
 C EnZD2 & ~z2Locked_S
 D ~E
 E EnPS & ~DonePS_S
 F ~H & ~EnZD2
 G ~H & EnZD2
 H EnZD1 & ~z1Locked_S
 I ~J
 J ~TO & ~FREEZEDLL_S
 K TO & ~FREEZEDLL_S & ~EnPS & EnZD1
 L TO & ~FREEZEDLL_S & EnPS
 M FREEZEDLL_S | (TO & ~EnPS & ~EnZD1)
 N ~TO & ~FREEZEDLL_S

O ~P
 P ~TO | FREEZEDLL_S
 Q DonePS_S & EnZD1
 R DonePS_S & ~EnZD1
 S ~DonePS_S
 T ~U
 U ~TO | FREEZEDLL_S
 V TO & ~FREEZEDLL_S & EnZD2
 W FREEZEDLL_S
 X ~TO & ~FREEZEDLL_S

State Machine